Deep Reinforcement Learning for Real-time Assembly Planning in Robot-based Prefabricated Construction

Aiyu Zhu, Tianhong Dai, Gangyan Xu, Pieter Pauwels, Bauke de Vries, Meng Fang

***Abstract*—The adoption of robotics is promising to improve the efficiency, quality, and safety of prefabricated construction. Besides technologies that improve the capability of a single robot, the automated assembly planning for robots at construction sites is vital for further improving the efficiency and promoting robots into practices. However, considering the highly dynamic and uncertain nature of a construction environment, and the varied scenarios in different construction sites, it is always challenging to make appropriate and up-to-date assembly plans. Therefore, this paper proposes a Deep Reinforcement Learning (DRL) based method for automated assembly planning in robot-based prefabricated construction. Specifically, a re-configurable simulator for assembly planning is developed based on a Building Information Model (BIM) and an open game engine, which could support the training and testing of various optimization methods. Furthermore, the assembly planning problem is modelled as a Markov Decision Process (MDP) and a set of DRL algorithms are developed and trained using the simulator. Finally, experimental case studies in four typical scenarios are conducted, and the performance of our proposed methods have been verified, which can also serve as benchmarks for future research works within the community of automated construction.**

Note to Practitioners: This paper is conducted based on the comprehensive analysis of real-life assembly planning processes in prefabricated construction, and the methods proposed could bring many benefits to practitioners. Firstly, the proposed simulator could be easily re-configured to simulate diverse scenarios, which can be used to evaluate and verify the operations' optimization methods and new construction technologies. Secondly, the proposed DRL-based optimization methods can be directly adopted in various robot-based construction scenarios, and can also be tailored to support the assembly planning in traditional human-based or human-robot construction environments. Thirdly, the proposed DRL methods and their performance in the four typical scenarios can serve as benchmarks for proposing new advanced construction technologies and optimization methods in assembly planning.

***Index Terms*—Prefabricated Construction, Assembly Planning, Deep Reinforcement Learning (DRL), Automated Construction, Building Information Modelling (BIM)**

This work is supported by China Scholarship Council (No.202007720036) and the National Natural Science Foundation of China (72174042). Aiyu Zhu, Tianhong Dai and Gangyan Xu contribute equally to this paper. (Corresponding Authors: Aiyu Zhu; Meng Fang)

This paper is an extension of Zhu et al. [1].

A. Zhu, P. Pauwels, Bd. Vries are with the Department of Information Systems in the Built Environment, Eindhoven University of Technology. E-mail: a.zhu@tue.nl, p.pauwels@tue.nl, b.d.vries@tue.nl

T. Dai is with Department of Computing Science, University of Aberdeen. Email: tianhong.dai@abdn.ac.uk

G. Xu is with Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. Email: gangyan.xu@polyu.edu.hk

M. Fang is with Department of Computer Science, University of Liverpool. Email: Meng.Fang@liverpool.ac.uk

I. Introduction

PREFABRICATED construction has been widely adopted around the world as it could bring many benefits including high efficiency, low cost, improved quality, and reduced waste [2]. Furthermore, along with the introduction of construction 4.0 [3] and construction servitization [4], many efforts have been made on the digitization and automation of prefabricated construction, from prefab production [5], prefabrication transportation [6], to on-site assembly [7]. However, the practices of automated prefabricated construction are still far from satisfactory, especially for the on-site assembly part. According to our investigations in Guangdong-Hong Kong-Macao Greater Bay Area, manual works and traditional decision-making methods based on expert knowledge still dominate the on-site assembly processes, which have been identified as major obstructions for fully reaping the benefits of prefabricated construction and further improving efficiency in construction planning and execution.

In recent years, along with the development of autonomous systems and tool-sets of Artificial Intelligence (AI), there is a trend of adopting robots in construction systems to improve the efficiency, quality, and safety of on-site assembly [8]–[11]. Many works have been done on designing new types of robots for specific tasks in diverse scenarios [12], automation technologies [13], decision-making models [14], [15], and human-robot collaboration strategies [16]–[18]. However, automated and real-time assembly planning methods for robots in prefabricated construction have been rarely studied as the foundation of improving the construction efficiency and lowering the cost.

The on-site assembly planning for robot-based prefabricated construction is complex and challenging. Firstly, it requires seamless cooperation among resources, equipment, and tasks with multiple objectives (e.g. efficiency, quality, and safety) and strict constraints (e.g. sequence of assembly and obstacle-free assembly paths), which are difficult to model. Secondly, the assembly processes vary a lot among different construction scenarios in terms of construction site layouts, construction tasks, prefab types, and the number of robots, which often leads to time-consuming and difficult to develop scenario-specific models for assembly planning. Thirdly, the assembly processes are highly dynamic with many uncertainties, such as availability of prefab elements and changing readiness states of robots. These uncertainties require the planning of decisions to be nearly real-time.

Recent advances in DRL provide promising opportunities to address the above challenges, and this technology has been successfully adopted to solve complex and dynamic optimization and control problems, such as the Vehicle Routing Problem (VRP) [19], traffic signal control [20], job shop scheduling [21], and multi-robot systems [22]. However, DRL is still seldom used to solve the planning issues for on-site prefab assembly, especially under the robot-based scenarios. Several questions remain open for discussion: (1) How to model the on-site assembly processes and build a flexible simulation environment for diverse prefabricated construction scenarios? (2) How to develop the optimization model and corresponding DRL-based assembly planning method to generate effective solutions? and (3) to what extent is the DRL-based method superior to traditional methods?

To address the above questions, this paper develops a re-configurable assembly simulator and proposes DRL-based methods to realize real-time assembly planning for robot-based prefabricated construction. Specifically, through modelling the basic elements and common activities in diverse prefabricated construction scenarios, a flexible simulator is developed based on Building Information Modelling (BIM) technology and an open game engine. It could be configured for various scenarios, simulate fine-grained actions of construction robots, and support the training and testing of optimization algorithms. Furthermore, the assembly planning problem is modelled as a Markov Decision Process (MDP) and a set of DRL algorithms are developed and trained to realize real-time decision-making of assembly planning, which could well cope with the dynamics of on-site assembly processes. Furthermore, experimental case studies are conducted in four scenarios to evaluate the effectiveness of the proposed methods, whose performance can serve as standardized benchmarks within the community of automated construction.

The contributions of this paper lie in the following three aspects:

1) This work builds the computational environment for adopting DRL in on-site operations of prefabricated construction, which provides a flexible test-bed and bridges the gap between DRL and decision-making in construction.
2) This work proposes planning methods based on DRL, which could well cope with scenario complexity and dynamics and be applied in many other fields.
3) This work provides a set of benchmarks for assembly planning in robot-based prefabricated construction, which could facilitate the design and evaluation of advanced planning approaches.

The remainder of this paper is structured as follows. Section II reviews relevant literature, and Section III presents the design of the simulation environment. Section IV models the on-site assembly planning problem and explains the DRL-based methods. Section V gives the experimental case studies and evaluates the performance of the proposed methods. Section VI concludes the whole paper and points out future research directions.

II. Literature Review

A. Assembly Planning in Prefabricated Construction

With the emergence of prefabricated construction, many efforts have been made on its assembly planning methods. As early as 2002, Balaguer et al. [23] proposed an approach to automate the assembly of future houses, in which transfer, positioning and assembly were treated as the main steps of onsite assembly. Through Geometric Reasoning, Hu [24] presented a component-based automated assembly method for prefabricated buildings. Žaková et al. [25] attempts to use construction knowledge as input to automate the planning of the construction process through an ontological reasoning approach. To ensure construction safety, Rahman [26] proposes a construction planning method with safety assessment. With the Smart Construction Objects (SCO) proposed in Niu et al. [27], an increasing number of studies have been implemented to control and plan the assembly process based on SCOs [8], [28], [29].

As one of the most significant tasks in prefabricated construction, crane-related assembly planning requires path generation from the yard location to the target location and finds the shortest path under the premise of collision-free movement [30]. Early works on path planning for construction boomed in the early 2000s [31], [32]. Several studies attempted to achieve near real-time path planning during the assembly process [33], [34]. However, few studies considered both crane operation efficiency and collision effects to improve path planning and re-planning prior to actual construction [35].

B. BIM-enabled Smart Construction

Building Information Modelling (BIM) is a set of tools and technologies to generate and manage the digital representations of physical and functional features of buildings [36]. Using BIM, the building can be represented with rich and accurate three-dimensional information [37], and could better support the decision-making processes in construction [38]. As a key technology for the digitization of buildings, BIM leads to an increase of smart construction research [39], which has been adopted in nearly every phase of construction [40]–[44].

In prefabricated construction, Li et al. [7] realized monitoring and management of on-site assembly based on BIM and Internet of Things (IoT). An et al. [45] proposed a BIM-enable design framework to improve the production and assembly efficiency of prefab components. Furthermore, the introduction of construction robots [46] and its integration with BIM further simulate the development of smart construction. For example, Gambao et al. [47] propose a robotic solution for assembly work in construction. By using the BIM model, Moura [48] proposes a method for localizing and modeling mobile robots at construction sites. Through the integration of BIM and robotic systems, Follini [49] proposed a method for multi-robot collaboration in building construction and maintenance. Considering the robot-based assembly process of hospitalisation facilities using prefab components, Gao et al. [50] developed a set of task and motion planning algorithms upon BIM-based prototypes. And Chong et al. [51] developed a simulation framework that integrates BIM and robotics for construction automation, and built a tool to make operational analysis based on BIM data.

Focusing on the assembly planning in various construction scenarios, Lu and Olofsson [52] designed a framework to simulate the construction process, which integrated BIM with discrete event processing models. Besides, based on BIM, Bortolini [53] presented a site logistics planning for prefabricated construction and Ding [54] proposed a task planning method for robotic brick assembly. Meanwhile, 3D planning of construction using BIM data is widely adopted in industry with tools such as Synchro, Primavera, and Microsoft Project.

C. DRL Applications

DRL has attracted a lot of attention from diverse real-world applications including games [55]–[58], robotics [59]–[62], transportation [19], [20], construction [63], healthcare [64], navigation [65], [66], and etc [67]. These studies are grounded in domain knowledge and use DRL as a method to solve optimization and control problems under complex and dynamic environments. For example, in the area of robot control, James [68] and Mahmood [69] propose a benchmark for robot learning to solving the problem of robot control planning. Hou [70] presents DRL methods for solving the assembly problem, and Gu et al. [71] achieved optimized robotic manipulator control using DRL. And Victor et al. [72] explores DRL for real-world autonomous systems, providing ideas for applying DRL to path planning systems for real applications. Although DRL has demonstrated its potential and effectiveness on solving practical problems in various fields, its application in construction industry is still rare, leaving many open opportunities.

III. Reconfigurable simulator for Assembly Operations

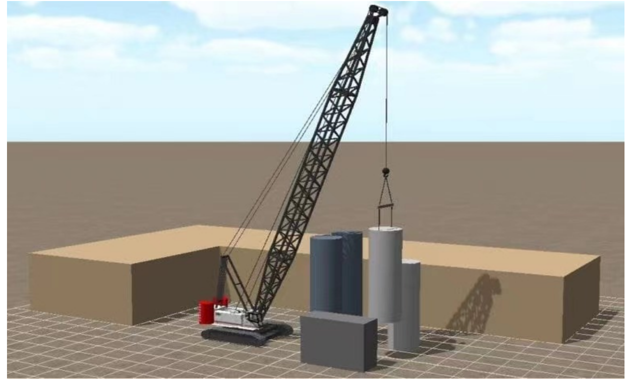
In order to investigate the use of DRL for real-time construction planning, a flexible and simple simulation environment for robot-enabled prefabricated construction is developed, as illustrated in Figure 1.

A. Scenario Description

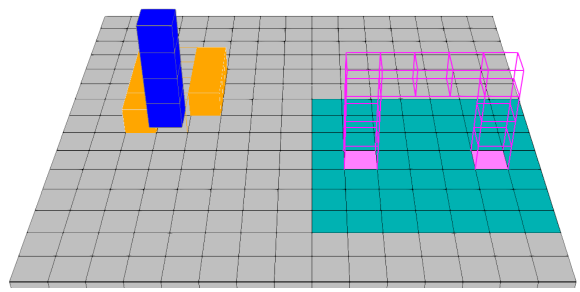
In prefabricated construction, a building is constructed by assembling individual prefab components together according to the construction plan. The assembly process consists of three steps. Firstly, component are transferred from the yard to the designed position in the building. Secondly, each connection node of each component is positioned to ensure the accurate connection location. And thirdly, the prefab components are connected (also called fixing the assembly).

To facilitate the adoption of robots for prefab components assembly, the construction tasks of each prefab component in above three steps are defined as states of the component. Then the robot can identify the current status of components based on their states, and take actions accordingly. The process is depicted as follows:

1) When a component is in the yard, its state is set as 'initial state'. It requires the robot to perform the task of transporting the component from yard to the build location.
2) When the component is being transported, its state is set to 'in transit' and no other robot is needed.
3) When the component arrives at the build position, its state is set to 'arrived', which requires a robot to perform the assembly task.
4) When the component is assembled, its state is set to 'assembled', which indicates that the assembly of the component is completed.



(a) Assembly work in construction



(b) Assembly task in our simplified simulation

Fig. 1: Overview of our construction simulations

Meanwhile, two sets of requirements should be considered when designing the simulator.

Safety requirements. In the practice of prefabricated construction, collision of prefab components during delivery is one of the main causes of engineering accidents. Thus collision of components is not allowed in the simulator. Furthermore, a minimum distance requirement between the construction area and the yard area is always required to ensure the safety of prefab components and workers. Therefore, in the simulator, the yard cannot be located within the construction area. Furthermore, in practice, all the construction activities should be performed in the designated area, so all the prefab components in the simulator cannot be moved or assembled outside given areas.

Assembly requirements. According to the basic rules in prefabricated construction, vertical components (columns) should be erected before horizontal components (beams). Therefore, each component will be checked for compliance with the construction requirements before it is built. Also, in practice, workers will adjust the orientation of the components before the components are lifted to improve the assembly efficiency, which will also be considered in the simulator.

Finally, six actions in three-dimensional (3D) space are designed to simulate the movement of components during the transfer by the tower crane. These six actions are up, down, left, right, forward, and backward. Each action moves the component one unit distance in the specified direction and each component can only perform one of these six movements at one time.

B. Simulator Development

Common construction simulation software (e.g. Synchro) typically focuses on visual simulation, using planned Gantt charts for simulation or manual modeling of the construction process with 3D visualisation features. These tools usually lack automatic planning capabilities and cannot provide interfaces for training RL policies. Therefore, a new simulation environment for DRL-based assembly planning is developed in this work, as illustrated in Figure 1. To describe the component-oriented prefabricated construction assembly process, we use 3D grids to simulate the construction environment and a simplified BIM model to represent the prefab components, and the focus is put on the construction tasks, assembly processes, and the DRL performance evaluation. Meanwhile, each component has independent construction requirements, and the assembly policy is planned under the constraints of environments and rules of construction.

The construction environment in our simulator is shown in Figure 2. The environment consists of three parts: component entities, construction site, and target. There are two types of component entities, namely, columns and beams; the area of the site is also divided into two types, they are yard and construction area, to comply with the safety requirements; and the construction target contains the target positions of prefab components.

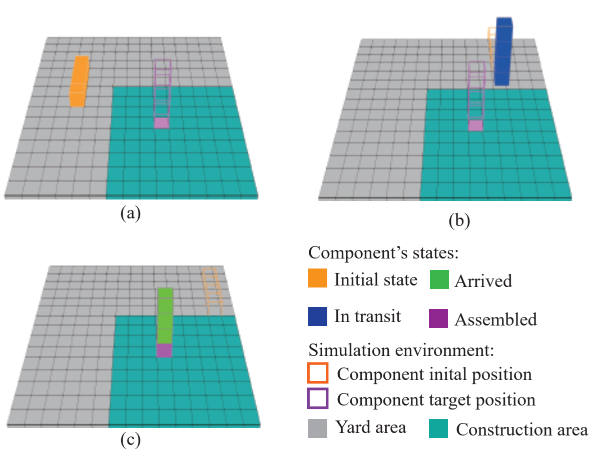


Fig. 2: Environment settings in construction simulations.

Based on the above analysis, the simulator is developed using pygame[1] and pyOpenGL[2]. In the simulator, the basic unit is a $1 \times 1 \times 1$ cube (dimensionless voxel). Each cube is defined by its spatial information as well as a number of state attributes. Based on the basic unit cube, a construction environment can be built with $X \times Y \times Z$ cubes (length $\times$ width $\times$ height).

As illustrated in Figure 2, a gray cube-composite area is a construction yard and a cyan cube-composite area is a construction area. Yard and construction area size are represented by the number of cubes. For example, if the length of a yard is 15 and the width is 10, then the yard consists of 15 $\times$ 10 cubes. Similarly, the components are formed by basic unit cubes. For example, the orange cube in Figure 2 (a) is a column with length 4. Since each component has four states, various colors are used to indicate the different states of the components to show the changes in the components during the construction process. Specifically, as shown in Figure 2 (a), orange is used to indicate the 'initial state' of the component, as shown in Figure 2 (b), blue is used to indicate the 'in transit' of the component, as shown in Figure 2 (c), green is used to indicate 'arrived' of the component, and pink to indicate that the nodes of the component are 'assembled'. Also, to better show the target of the construction, we use orange wireframe to indicate the initial position of the component and pink wireframe to indicate the target position of the component.

The simulator requires the user to initialize the following information:

1) Site information: the user needs to set the dimensions of the construction environment, construction area and yard area, respectively.
2) Component information: the user needs to declare the number of components, the type of each component (column or beam), the dimensions, and the starting and target positions of components.

In practice, the target position of components should be fixed and can be imported from the real BIM model, while their initial positions still need to be configured by users. Such information (initial position) can be retrieved using 4D BIM planning tools that include a full construction procedure. Taking this into account, the remainder of this work focuses on the execution of the DRL policy itself.

IV. DRL for Assembly Planning

In this section, the DRL methods for assembly planning will be discussed based on the proposed simulator.

A. Preliminary

a) Markov Decision Process (MDP): In RL, an agent interacts with an environment that is modeled as an MDP [73]. It can be represented as (S, A, P, T, R, γ), where S is the set of states, A is the set of actions, and P is the initial state distribution. $T(s_{t+1}|s_t, a_t)$ is the probability of transitioning from state s_t to $s_{t+1}, s_t, s_{t+1} \in S$ when action $a_t \in A$ is conducted, $R(r_{t+1}|s_t, a_t)$ is the probability of receiving

[1]https://www.pygame.org/news
[2]http://pyopengl.sourceforge.net/

reward $r_{t+1} \in R$ after executing action a_t in state s_t, and $\gamma \in [0, 1)$ is the discount factor.

b) RL algorithms: Value-based RL methods aim to learn the value function of each state or state-action pair of the optimal policy π. The state value function for a particular policy π can be denoted as $V_\pi(s), \forall s \in S$ while the state-action value function is denoted $Q_\pi(s, a), \forall s, a \in (S, A)$. In order to find the value functions corresponding to the optimal policy π^*, the functions can be updated as : $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$. The optimal policy is found by acting greedily over the optimal value function at each state $\pi^*(s) = \arg\max_a Q^*(s, a)$. Policy-based methods are methods that directly learn the policy as a parameterized function π_θ rather than learn the value function explicitly, where the parameters of the function are θ. Policy gradients [74] use the update function: $\theta_{t+1} = \theta_t + \alpha(G_t - b(S_t))\frac{\nabla_\theta \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$, where α is the step size, $b(S_t)$ is a baseline, and the return is $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$. Actor-Critic (A2C) methods are hybrid value-based and policy-based methods that directly learn both the policy (actor) and the value function (critic) [75]. The update function for actor-critic is: $\theta_{t+1} \leftarrow \theta_t + \alpha(R_{t+1} + \gamma\hat{V}(S_{t+1}) - \hat{V}(S_t))\frac{\nabla_\theta \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)}$, where $\hat{V}(.)$ is a parameterized estimate of the optimal value function.

B. DRL Model for assembly planning

After the construction assembly process is transformed into RL executable states, actions and rewards, the simulation environment can be connected with RL algorithms for generating assembly planning policies. Figure 3 shows the framework of the simulation environment interacting with the RL agent. In this framework, simulation data will be converted as RL agent observable structure and then sent to the RL agent. Based on the input, the RL agent will select an action using the current policy. The action is evaluated by the reward feedback from the simulation, and the construction environment will be updated and passed to the RL agent again until the assembly process ends.

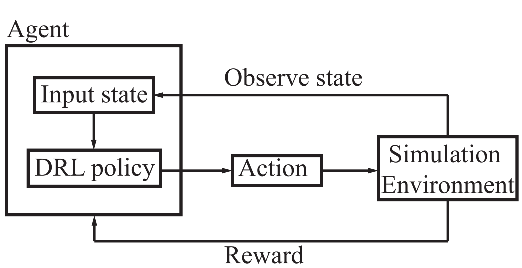


Fig. 3: Framework of DRL for Assembly Planning.

The assembly planning process of prefabricated construction is modelled as MDP. In the following, the states, actions, and rewards in the MDP model will be discussed in more detail.

States

The states or observations of a construction environment are defined as a tensor of shape $W * L * d$, where W indicates the width and L indicates the length of the environment. The variable d refers to a 6-tuple that encodes the features and status of each cube using 6 integer values:

1) a scalar indicates the height (Z_i) of the component on the yard;
2) a scalar (0 or 1) indicates whether the number of transit steps of the building component is more than a predefined maximum step, if there is a building component in that location;
3) a scalar indicates the number of steps that have been moved;
4) a scalar represents the height (Z_t) of the component;
5) a scalar indicates the height (Z_{tar}) of the component's target position if it belongs to the target, or else 0;
6) a scalar expresses the number of components that have completed the planning.

Actions

An action is an operation on a component by a crane. We define that a component can be executed in six directions: forward, backward, left, right, up, and down. Then we have six actions $a \in [0, 1, 2, 3, 4, 5]$, where each number indicates an operation with a direction.

Rewards

The design of rewards is one of the critical factors affecting the success and efficiency of learning. In practice, construction projects are often expected to be completed in the fastest yet most reasonable amount of time. In the simulation environment, the time factor is transformed into the number of steps to finish the assembly. Thus, the objective is to achieve the construction task with as few steps (t) as possible. According to the above requirements, our design for the reward structure is as following:

1) To minimize the number of component actions, we set the agent to receive a basic reward (r_0, and $r_0 = -1$ in practice) for every step (action). Based on this design, the agent needs to finish the construction with the least number of steps (t) to maximize the total reward;
2) To avoid over-exploration by the agent in learning the construction of individual components, we impose a limit on the number of steps each component can take to build. Here, we set a threshold (ε) for the maximum number of steps (t), and when the execution steps of a single component are larger than the threshold, each step executed after that component will be subject to a double basic reward ($r_0 = -2$);
3) When each component is well constructed, the agent will receive a positive reward, and when all components are well constructed, the agent will receive a larger positive reward;
4) A progressive reward mechanism is used to guide the agent to complete the construction task, that is, the more components are built, the agent will receive more rewards.

The reward is summarized as following:

$$R = \sum_{i=1}^{C} \sum_{t=0}^{T} R_{c_i,t} \tag{1}$$

where $R_{c_i,t}$ represents the rewards of a single component. i is the index of the component and t is the total step of the component.

$R_{c_i,t}$ is defined as:

$$R_{c_i,t} = \begin{cases} t * \frac{r_0}{c_i} \quad + i * c_i, & t \leq \varepsilon \\ \varepsilon * \frac{r_0}{c_i} \quad + 2 * (t - \varepsilon) * \frac{r_0}{c_i} \quad + i * c_i, & t > \varepsilon \end{cases} \tag{2}$$

where c_i represents the serial number of the component during the assembly process, i represents the index of the component.

For example, the first component to be constructed is marked as c_1 with serial number 1, and the n-th component to be assembled is marked as c_n with serial number n. t is the number of steps for each component. When the component is not moving, t is 0; when the component starts moving, t is the number of the steps of the component. ε is the step threshold for each component. n is a constant that can be specified as a positive integer as the number of components in the environment increases, and we use it to increase the reward each component receives for completing the assembly and to decrease the penalty the component receives for each step it moves.

V. Experimental Case Study

In this section, an experimental case study with four scenarios that are commonly used in the practice of prefabricated construction is conducted to verify the performance of the proposed simulator and methods.

A. Scenarios

We consider four typical scenarios that are widely adopted in the practice of prefabricated construction, as illustrated in Figure 4. Based on these four real-life scenarios, four corresponding simulation scenarios were developed, as shown in Figure 5.

a) Scenario 1: Assembly planning with dynamic site layout: In prefabricated construction, although the layout of the construction site is pre-determined, it can in very rare cases be changed due to the dynamics of weather, design requirements, and schedule modification. This happens rarely, as contractors aim to stick to the plan. Yet, in this case, the supply point of components will be re-located to satisfy the new layout [35], and the new supply point requires a new assembly path. As the scenario [76] in Figure 4a shows, the prefabricated wall panel is unique: its location in the building is determined and the components need to be transferred from the yard location to their unique location in the building. However, in a dynamic layout scenario, the initial location can be anywhere in the yard, as shown in Figure 5a.

b) Scenario 2: Assembly planning with interchangeable standardized components: With the development of prefabricated buildings and the requirement of sustainable construction, interchangeable standardized components are gradually used in actual construction. The recently released ISO 20887[3] also promotes designing interchangeable and standard size components [77]. It should be noted that this type of component is mostly used in steel-based prefabrication but has not been widely used in prefabricated concrete construction, especially in pre-assembly scenarios for off-site modular housing or on-site light steel housing. Standardized components are usually stored in the required locations, but their target locations may be multiple [78], which means they can be assembled in different locations in the structure. For example, as shown in Figure 4b [79], the standardized truss is interchangeable, so it does not need to be assembled at a unique location in the structure. Any location that satisfies the structural requirements can be used as the target location for this standardized component. Figure 5b shows the simulation environment of scenario 2. The initial positions of components are fixed and we allow the target positions of the same standardized components to be interchangeable. For example, the target position of Column 1 can be any of the columns' targets.

c) Scenario 3: Assembly planning with dynamic site layout and interchangeable standardized components: Scenarios 1 and 2 describe assembly processes that limit only the initial or target locations, while construction sites are often more complex, and both the initial and target locations of components can be changed to improve construction efficiency and safety [80]. In particular, many steel structures are of essentially the same design, but when similar projects are built on different sites, the location of the components on the site may change depending on the site environment while the same size and type of components can be placed in any location that conforms to the structural requirements. For example, in Figure 4c [81], there are many buildings of the same structure. During the construction processes, the location of the components in the yard changes and the same types of these components can be assembled at any location that meets structural requirements. Scenario 3 is shown in Figure 5c. The initial positions of components are random, while their target positions are interchangeable. For example, the initial position of Column 1 will be randomly placed in the field, and its target position can also be any of the columns' targets.

d) Scenario 4: Assembly planning in crowded spaces with obstacles: Construction sites are always complex environments that involve many components and equipment, and should consider both efficiency and safety. Collisions should therefore be avoided [82], especially in the construction sites located in the urban areas, as shown in Figure 4d [83]. Therefore, in Scenario 4 shown in Figure 5d, some obstacles are added to simulate the potential collisions in the construction processes. The black wire-frame indicates other buildings in the construction site, which are obstacles under construction.

B. Environments

In addition to the four scenarios given in Fig. 5, four environments are designed for each scenario. The main differences between these environments are the different target construction buildings, from simple to complex. This is mainly used to simulate the planning of different levels of complexity of structures in the above construction scenarios. The complexity

[3]https://www.iso.org/standard/69370.html

(a) Assembly planning with dynamic site layout

(b) Assembly planning with interchangeable standardized components

(c) Assembly planning with dynamic site layout and interchangeable standardized components

(d) Assembly planning in crowded spaces with obstacles

Fig. 4: Realistic applications correspondence of our scenarios.

of the structure here is reflected in the number of components, and the increase in the number of components to be planned indicates a more complex structure as displayed in in Figure 6.

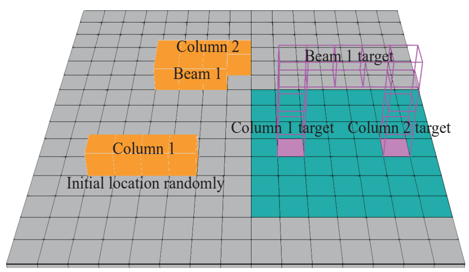
The total construction environment measures 15 $\times$ 15 units in length and width and 8 units in height where we use x, y, and z to represent the length, width, and height of the site, respectively. Construction areas are different for different environments (blue areas): Envs 1 and 2 are $x \in (9, 15)$ and $y \in (6, 13)$; Env 3 is $x \in (8, 15)$ and $y \in (8, 15)$; Env 4 is $x \in (8, 15)$ and $y \in (5, 15)$. The length of the beam is 3 units, the length of the column is 4 units. The settings for each environment are as follows.

- Env1: The target structure is made up of two columns and one beam, and one column has been built. The controller needs to plan for the remaining one column and one beam;
- Env2: The target structure is identical to the target structure in Env1, except that the controller needs to plan for two columns and a beam;
- Env3: The target structure consists of four columns and four beams. The controller needs to plan for eight components;
- Env4: he target structure has more components, namely six columns and seven beams. The controller needs to plan for all thirteen components.

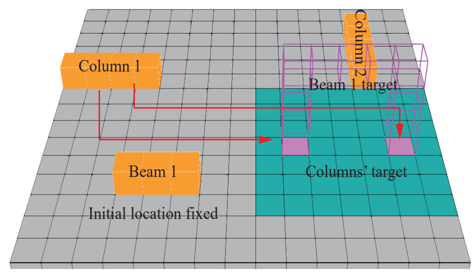
C. Setup and Candidate Controllers

There are 4×4 situations in total (4 environments; 4 scenarios), and in our experiment, controllers are set up to learn each of these 16 situations. They are also evaluated separately for each situation. The evaluation results are plotted in Figure 7 based on repeated runs with 15 different seeds. For each of the 16 situations, the 4 following candidate controllers are evaluated for the DRL; and the four controllers are plotted in Figure 7. The solid lines are the mean rewards over 10 episodes, and the shaded area represents the corresponding standard errors in Figure 7. The hyperparameters and training details are given in the appendix.

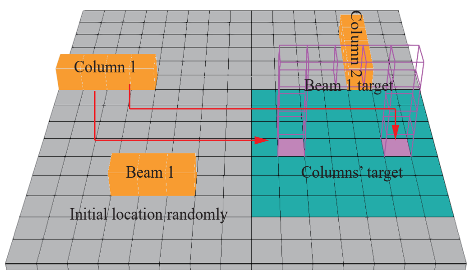
- DQN [84]: The vanilla deep Q network (DQN), which is an off-policy RL algorithm;
- DDQN [85]: Double DQN (DDQN) employs two Q-value functions to reduce the over-estimation problem in the original DQN algorithm;
- A2C [86]: A2C utilizes multiple CPUs to collect experiences from separate environments to reduce the correlations between samples and enable faster learning;
- PPO [59]: PPO adopts a clip function to restrict the change of policy in each update which leads to a stable training.



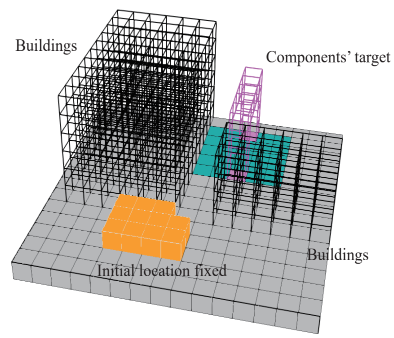
(a) Assembly planning with dynamic site layout



(b) Assembly planning with interchangeable standardized components

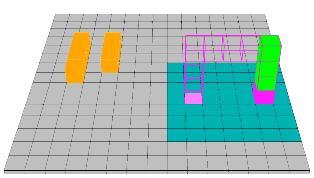


(c) Assembly planning with dynamic site layout and interchangeable standardized components

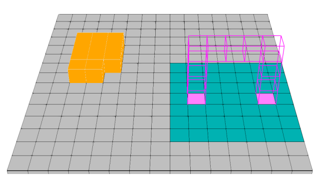


(d) Assembly planning in crowded spaces with obstacles

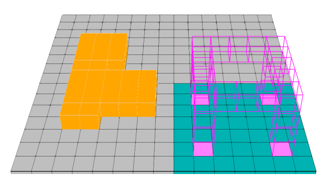
Fig. 5: Benchmarks based on four scenarios.



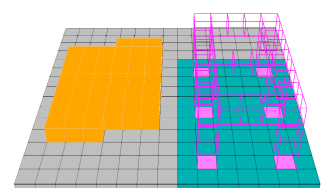
(a) Env1



(b) Env2



(c) Env3



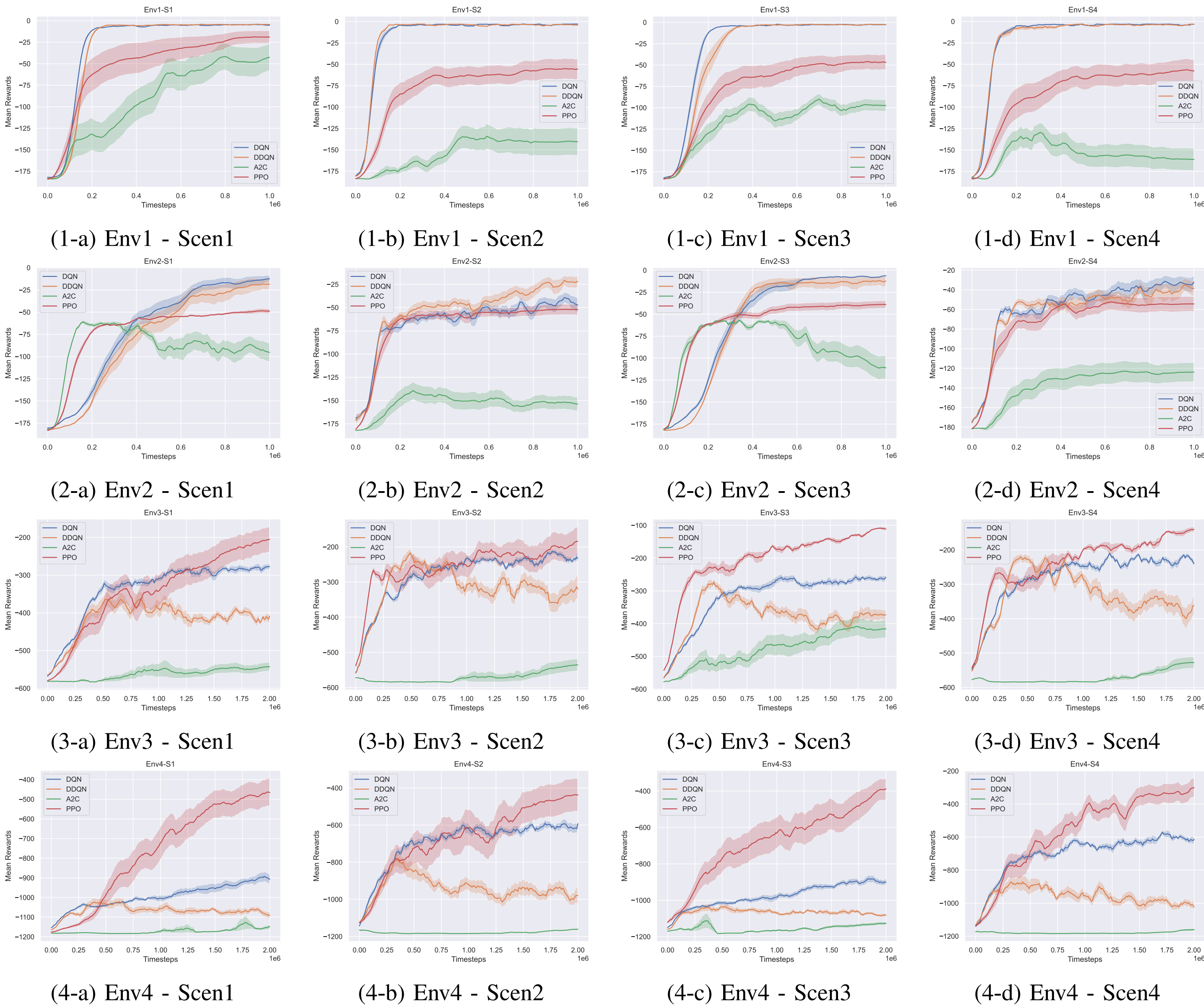
(d) Env4

Fig. 6: Four environments, ranging from simple to more complex.

D. Results

Table I shows the resulting average return/reward values and their standard errors for each of the DRL algorithms and environments. As can be seen in this Table, in the first two environments, which only have 2 and 3 components to manipulate (Env1 and Env2), the value-based algorithms (DQN and DDQN) perform better than policy-based methods. More specifically, in Env1, DQN and DDQN achieve similar performance. In Env2, DQN outperforms DDQN in 3 out of 4 scenarios. For policy-based methods, PPO outperforms A2C in all scenarios. In Env3 and Env4, which have 8 and 13 components to manipulate, PPO achieves the best performance among all methods. DQN has better performance than DDQN. A2C has the worst performance in all scenarios. In general, PPO achieves better performance in more complex environments. This is because PPO utilises a clipped surrogate objective to constrain the variation of policy in each update, which could prevent the performance drop during training. In addition, PPO is more robust to the hyperparameter setting.

Furthermore, Figure 7 presents the learning curve of each of the controller algorithms. In Env1, value-based methods (DQN and DDQN) converge to higher performance quickly (average reward is around -4.). On the other hand, policy-based methods (A2C and PPO) converge to a relatively low performance. In Env2, PPO has a faster or similar convergence rate than value-based methods at the beginning of training. However, value-based method perform better at the end of training. In Env3 and Env4, the performance of PPO increases with increasing number of timesteps and outperforms other baselines in all scenarios. DQN also demonstrates this tendency and achieves



(1-a) Env1 - Scen1 (1-b) Env1 - Scen2 (1-c) Env1 - Scen3 (1-d) Env1 - Scen4

(2-a) Env2 - Scen1 (2-b) Env2 - Scen2 (2-c) Env2 - Scen3 (2-d) Env2 - Scen4

(3-a) Env3 - Scen1 (3-b) Env3 - Scen2 (3-c) Env3 - Scen3 (3-d) Env3 - Scen4

(4-a) Env4 - Scen1 (4-b) Env4 - Scen2 (4-c) Env4 - Scen3 (4-d) Env4 - Scen4

Fig. 7: Comparison between RL algorithms in all environments with different scenarios.

the second best performance in all scenarios. DDQN has a tendency to improve performance at the beginning. However, its performance starts to decrease at around 0.5 million timesteps. A2C has the worst performance in all environments.

As far as these simplified environments are concerned, RL provides helpful assistance in the problem of assembly planning during the construction process. For example, in Scenario 1, when the initial position of a component changes, the component can still rapidly output a legal path based on trained deep neural models, avoiding recalculation of the path planning. The RL algorithm can be well employed in path planning along with collision analysis, which solves shortest path and collision-free planning at the same time.

VI. Discussion

According to the experimental results presented above, the proposed methods work well in different cases and could learn the optimal assembly policies efficiently, which provides new perspectives on the assembly planning in prefabricated construction. Currently, we experiment with the application of the DRL algorithm to the construction planning problem through a simplified simulation. As some of our benchmark results illustrate, in certain cases, the agent learns construction policies, while in others, the agent struggles to learn an optimal policy as the number of components grows. So, in the present scenarios with fewer components, our framework enables agents to carry out assembly planning using construction rules, which offers a fresh perspective on the assembly planning problem.

However, there are also some limitations. On one hand, the proposed methods are evaluated in simplified simulation scenarios with relatively limited types of components. In future research, it needs to be evaluated for its applicability and robustness in combination with realistic BIM models and realistic construction schedules (4D BIM and construction logs). Such evaluation particularly needs to investigate opportunities for making this work better scalable.

Furthermore, it needs to be mentioned that the assembly

TABLE I: Mean rewards ± standard errors over 15 random seeds in the last 100 episodes on all environments. The best results are in **bold**.

	DQN	DDQN	A2C	PPO
E1S1	-5.16±0.49	**-4.27±0.42**	-42.57±15.60	-19.03±7.34
E1S2	**-2.94±0.14**	-4.36±0.99	-140.61±16.07	-55.46±12.13
E1S3	**-2.62±0.30**	-2.62±0.32	-97.42±7.01	-46.89±8.86
E1S4	-3.37±0.32	**-3.16±0.14**	-161.02±13.10	-57.80±13.09
E2S1	**-12.40±3.76**	-18.42±5.90	-95.45±10.49	-48.87±2.73
E2S2	-47.27±7.94	**-21.76±5.90**	-153.80±7.35	-52.17±5.67
E2S3	**-6.16±0.40**	-12.27±5.75	-110.95±13.17	-38.93±4.42
E2S4	**-32.36±7.09**	-38.63±7.83	-123.76±9.77	-54.40±7.64
E3S1	-277.28±10.29	-409.16±15.14	-543.08±11.76	**-204.94±33.17**
E3S2	-232.77±12.97	-319.96±34.05	-535.54±17.38	**-183.85±41.07**
E3S3	-258.80±8.06	-373.12±16.29	-415.12±28.28	**-111.22±6.13**
E3S4	-239.68±10.29	-362.18±28.83	-527.82±16.68	**140.18±7.48**
E4S1	-907.31±25.98	-1091.58±17.01	-1148.31±13.42	**-466.27±70.70**
E4S2	-593.04±28.46	-978.52±47.37	-1160.99±4.69	**-437.76±89.00**
E4S3	-900.72±15.95	-1080.51±9.15	-1127.31±9.10	**-388.21±58.61**
E4S4	-614.14±24.60	-1022.09±24.89	-1160.34±7.75	**-300.97±57.27**

planning policy needs to be re-trained whenever one changes to a new scenario. As this costs time and resources, it becomes undesirable to go through highly dynamic and complex construction sites that change scenarios too frequently. This is anyhow also desirable in any construction site in general.

VII. Conclusions and Future Work

In this paper, we have designed a simplified simulation environment to simulate robot-based construction processes and how they can be planned using DRL. In this simulation environment, we considered four construction scenarios and four construction environments (simple to complex). We used these 16 cases (4 times 4) to set up basic performance benchmarks in terms of available DRL-based controllers. Although these benchmarks are simplified for now, it is foreseen that (D)RL has great potential to improve construction planning in construction sites that are expected to be dynamic. Our results show that DQN and DDQN generally outperform the other DRL algorithms for our case scenarios and environments.

In forthcoming work, we will first and foremost look into (a) constructing more realistic construction sites that are built based on 4D planning software and real BIM models, which should contain more types and numbers of components; and (b) increasing performance of the promising DRL algorithms (DQN and DDQN) using an optimized process that at any given time takes into account only a reasonable number of components (short-sighted lean planner). This last improvement is expected to improve scalability of the simulation environment as well.

Appendix A
Experiment Setup

Hyperparameters and training details of baseline RL algorithms are given below:

- DQN and DDQN: batch size is 64, replay memory size is 50000, target network update frequency is 500, learning rate is 0.0005, initial exploration ratio is 1, final exploration ratio is 0.02, discount factor is 0.9 for Env1 and Env2, and 0.8 for Env3 and Env4.
- A2C: number of workers is 16, steps per worker in each rollout is 5, learning rate is 0.0007, discount factor is 0.9, GAE coefficient is 0.95, entropy coefficient is 0.01.
- PPO: number of workers is 16, batch size is 64, steps per worker in each rollout is 125, learning rate is 0.0003, discount factor is 0.9, GAE coefficient is 0.95, entropy coefficient is 0.1, update epochs is 4.

The code of this paper is publicly available at: https://github.com/hyintell/drl-assembly-planning.

References

[1] A. Zhu, G. Xu, P. Pauwels, B. de Vries, and M. Fang, "Deep reinforcement learning for prefab assembly planning in robot-based prefabricated construction," in *IEEE International Conference on Automation Science and Engineering*, 2021, pp. 1282–1288.

[2] G. Xu, M. Li, C.-H. Chen, and Y. Wei, "Cloud asset-enabled integrated iot platform for lean prefabricated construction," *Automation in Construction*, vol. 93, pp. 123–134, 2018.

[3] W. S. Alaloul, M. Liew, N. A. W. A. Zawawi, and I. B. Kennedy, "Industrial revolution 4.0 in the construction industry: Challenges and opportunities for stakeholders," *Ain Shams Engineering Journal*, vol. 11, no. 1, pp. 225–230, 2020.

[4] H. Wang and D. Liu, "Is servitization of construction the inevitable choice of internet plus construction?" *Frontiers of Engineering Management*, vol. 4, no. 2, pp. 229–230, 2017.

[5] K. Orlowski, "Automated manufacturing for timber-based panelised wall systems," *Automation in Construction*, vol. 109, p. 102988, 2020.

[6] G. Xu, M. Li, L. Luo, C.-H. Chen, and G. Q. Huang, "Cloud-based fleet management for prefabrication transportation," *Enterprise Information Systems*, vol. 13, no. 1, pp. 87–106, 2019.

[7] C. Z. Li, F. Xue, X. Li, J. Hong, and G. Q. Shen, "An internet of things-enabled bim platform for on-site assembly services in prefabricated construction," *Automation in construction*, vol. 89, pp. 146–161, 2018.

[8] A. Zhu, P. Pauwels, and B. De Vries, "Smart component-oriented method of construction robot coordination for prefabricated housing," *Automation in Construction*, vol. 129, p. 103778, 2021.

[9] C. P. Chea, Y. Bai, X. Pan, M. Arashpour, and Y. Xie, "An integrated review of automation and robotic technologies for structural prefabrication and construction," *Transportation Safety and Environment*, vol. 2, no. 2, pp. 81–96, 2020.

[10] C. Wang, W. Chi, Y. Sun, and M. Q.-H. Meng, "Autonomous robotic exploration by incremental road map construction," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1720–1731, 2019.

[11] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[12] F. Xu, S. Dai, Q. Jiang, and X. Wang, "Developing a climbing robot for repairing cables of cable-stayed bridges," *Automation in Construction*, vol. 129, p. 103807, 2021.

[13] I. Muhammad, K. H. Ying, M. S. Nithish, J. Y. Xin, Z. Xinge, and C. C. Cheah, "Robot-assisted object detection for construction automation: Data and information-driven approach," *IEEE/ASME Transactions on Mechatronics*, 2021.

[14] C. P. Schimanski, G. P. Monizza, C. Marcher, and D. T. Matt, "Development of a bim-based production planning and control system for lean construction through advancement and integration of existing management techniques," *Frontiers of Engineering Management*, pp. 1–13, 2020.

[15] A. K. Ali, O. J. Lee, and H. Song, "Robot-based facade spatial assembly optimization," *Journal of Building Engineering*, vol. 33, p. 101556, 2021.

[16] C.-J. Liang, X. Wang, V. R. Kamat, and C. C. Menassa, "Human–robot collaboration in construction: Classification and research trends," *Journal of Construction Engineering and Management*, vol. 147, no. 10, p. 03121006, 2021.

[17] Y. Liu, M. Habibnezhad, and H. Jebelli, "Brainwave-driven human-robot collaboration in construction," *Automation in Construction*, vol. 124, p. 103556, 2021.

[18] S. You, J.-H. Kim, S. Lee, V. Kamat, and L. P. Robert Jr, "Enhancing perceived safety in human–robot collaborative construction using immersive virtual environments," *Automation in Construction*, vol. 96, pp. 161–170, 2018.

[19] Y. Xu, M. Fang, L. Chen, G. Xu, Y. Du, and C. Zhang, "Reinforcement learning with multiple relational attention for solving vehicle routing problems," *IEEE Transactions on Cybernetics*, 2021.

[20] Y. Zhao, G. Xu, Y. Du, and M. Fang, "Learning multi-agent communication with policy fingerprints for adaptive traffic signal control," in *IEEE International Conference on Automation Science and Engineering*, 2020, pp. 266–273.

[21] S. Luo, L. Zhang, and Y. Fan, "Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, 2021.

[22] Z. Liu, Q. Liu, L. Tang, K. Jin, H. Wang, M. Liu, and H. Wang, "Visuomotor reinforcement learning for multirobot cooperative navigation," *IEEE Transactions on Automation Science and Engineering*, 2021.

[23] C. Balaguer, M. Abderrahim, J. Navarro, S. Boudjabeur, P. Aromaa, K. Kahkonen, S. Slavenburg, D. Seward, T. Bock, R. Wing, and B. Atkin, "Futurehome: An integrated construction automation approach," *IEEE Robotics Automation Magazine*, vol. 9, no. 1, pp. 55–66, 2002.

[24] W. Hu, "Automatic construction process of prefabricated buildings on geometric reasoning," in *Construction Research Congress 2005: Broadening Perspectives*, 2005, pp. 1–10.

[25] M. Žaková, P. Křemen, F. Železný, and N. Lavrač, "Automating knowledge discovery workflow composition through ontology-based planning," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 253–264, 2011.

[26] M. M. Rahman, L. Bobadilla, A. Mostafavi, T. Carmenate, and S. A. Zanlongo, "An automated methodology for worker path generation and safety assessment in construction projects," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 479–491, 2018.

[27] Y. Niu, W. Lu, K. Chen, G. G. Huang, and C. Anumba, "Smart construction objects," *Journal of Computing in Civil Engineering*, vol. 30, no. 4, p. 04015070, 2016.

[28] C. Rausch, M. Nahangi, M. Perreault, C. T. Haas, and J. West, "Optimum assembly planning for modular construction components," *Journal of Computing in Civil Engineering*, vol. 31, no. 1, p. 04016039, 2017.

[29] R. Y. Zhong, Y. Peng, F. Xue, J. Fang, W. Zou, H. Luo, S. T. Ng, W. Lu, G. Q. Shen, and G. Q. Huang, "Prefabricated construction enabled by the internet-of-things," *Automation in Construction*, vol. 76, pp. 59–70, 2017.

[30] Z. Zhang and W. Pan, "Lift planning and optimization in construction: A thirty-year review," *Automation in Construction*, vol. 118, p. 103271, 2020.

[31] H. R. Reddy and K. Varghese, "Automated path planning for mobile crane lifts," *Computer-Aided Civil and Infrastructure Engineering*, vol. 17, no. 6, pp. 439–448, 2002.

[32] S.-C. Kang and E. Miranda, "Computational methods for coordinating multiple construction cranes," *Journal of Computing in Civil Engineering*, vol. 22, no. 4, pp. 252–263, 2008.

[33] H. AlBahnassi and A. Hammad, "Near real-time motion planning and simulation of cranes in construction: Framework and system architecture," *Journal of Computing in Civil Engineering*, vol. 26, no. 1, pp. 54–63, 2012.

[34] W. Ren, Z. Wu, and L. Zhang, "Real-time planning of a lifting scheme in mobile crane mounted controllers," *Canadian Journal of Civil Engineering*, vol. 43, no. 6, pp. 542–552, 2016.

[35] C. Zhang and A. Hammad, "Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency," *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 396–410, 2012.

[36] R. Volk, J. Stengel, and F. Schultmann, "Building information modeling (bim) for existing buildings — literature review and future needs," *Automation in Construction*, vol. 38, no. 1, pp. 109–127, 2014.

[37] Costa and G., "Connecting building component catalogues with bim models using semantic technologies: an application for precast concrete components." *Automation in Construction*, vol. 57, pp. 239–248, 2015.

[38] L. Ding, Y. Zhou, and B. Akinci, "Building information modeling (bim) application framework: The process of expanding from 3d to computable nd," *Automation in Construction*, vol. 46, no. oct., pp. 82–93, 2014.

[39] J. Zhang, H. Luo, and J. Xu, "Towards fully bim-enabled building automation and robotics: A perspective of lifecycle information flow," *Computers in Industry*, vol. 135, p. 103570, 2022.

[40] Z. Ding, S. Liu, L. Liao, and L. Zhang, "A digital construction framework integrating building information modeling and reverse engineering technologies for renovation projects," *Automation in Construction*, vol. 102, pp. 45–58, 2019.

[41] R. Chahrour, M. A. Hafeez, A. M. Ahmad, H. I. Sulieman, H. Dawood, S. Rodriguez-Trejo, M. Kassem, K. K. Naji, and N. Dawood, "Cost-benefit analysis of bim-enabled design clash detection and resolution," *Construction Management and Economics*, vol. 39, no. 1, pp. 55–72, 2021.

[42] K. Whitlock, F. Abanda, M. Manjia, C. Pettang, and G. Nkeng, "Bim for construction site logistics management." *Journal of Engineering, Project, and Production Management*, vol. 8, no. 1, p. 47, 2018.

[43] Z. Yuan, C. Sun, and Y. Wang, "Design for manufacture and assembly-oriented parametric design of prefabricated buildings," *Automation in Construction*, vol. 88, pp. 13–22, 2018.

[44] R. Bortolini, C. T. Formoso, and D. D. Viana, "Site logistics planning and control for engineer-to-order prefabricated building systems using bim 4d modeling," *Automation in Construction*, vol. 98, pp. 248–264, 2019.

[45] S. An, P. Martinez, M. Al-Hussein, and R. Ahmad, "Automated verification of 3d manufacturability for steel frame assemblies," *Automation in Construction*, vol. 118, p. 103287, 2020.

[46] T. Bock, "Construction robotics," *Autonomous Robots*, vol. 22, no. 3, pp. 201–209, 2007.

[47] E. Gambao, C. Balaguer, and F. Gebhart, "Robot assembly system for computer-integrated construction," *Automation in Construction*, vol. 9, no. 5, pp. 479–487, 2000.

[48] M. S. Moura, C. Rizzo, and D. Serrano, "Bim-based localization and mapping for mobile robots in construction," in *IEEE International Conference on Autonomous Robot Systems and Competitions*, 2021, pp. 12–18.

[49] C. Follini, V. Magnago, K. Freitag, M. Terzer, C. Marcher, M. Riedl, A. Giusti, and D. T. Matt, "Bim-integrated collaborative robotics for application in building construction and maintenance," *Robotics*, vol. 10, no. 1, p. 2, 2020.

[50] Y. Gao, J. Meng, J. Shu, and Y. Liu, "Bim-based task and motion planning prototype for robotic assembly of covid-19 hospitalisation units—flatpack house," *Automation in Construction*, p. 104370, 2022.

[51] O. W. Chong, J. Zhang, R. M. Voyles, and B.-C. Min, "Bim-based simulation of construction robotics in the assembly process of wood frames," *Automation in Construction*, vol. 137, p. 104194, 2022.

[52] W. Lu and T. Olofsson, "Building information modeling and discrete event simulation: Towards an integrated framework," *Automation in Construction*, vol. 44, no. aug., pp. 73–83, 2014.

[53] R. Bortolini, C. T. Formoso, and D. D. Viana, "Site logistics planning and control for engineer-to-order prefabricated building systems using bim 4d modeling," *Automation in Construction*, vol. 98, no. FEB., pp. 248–264, 2019.

[54] L. Ding, W. Jiang, Y. Zhou, C. Zhou, and S. Liu, "Bim-based task-level planning for robotic brick assembly through image-based 3d modeling," *Advanced Engineering Informatics*, vol. 43, p. 100993, 2020.

[55] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[56] T. Tomilin, T. Dai, M. Fang, and M. Pechenizkiy, "Levdoom: A benchmark for generalization on level difficulty in reinforcement learning," in *IEEE Conference on Games*, 2022.

[57] Y. Du, L. Han, M. Fang, J. Liu, T. Dai, and D. Tao, "Liir: Learning individual intrinsic reward in multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 4403–4414, 2019.

[58] Y. Xu, M. Fang, L. Chen, Y. Du, J. T. Zhou, and C. Zhang, "Deep reinforcement learning with stacked hierarchical attention for text-based games," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 16 495–16 507.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[60] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[61] M. Fang, C. Zhou, B. Shi, B. Gong, J. Xu, and T. Zhang, "Dher: Hindsight experience replay for dynamic goals," in *International Conference on Learning Representations*, 2018.

[62] M. Fang, T. Zhou, Y. Du, L. Han, and Z. Zhang, "Curriculum-guided hindsight experience replay," *Advances in Neural Information Processing Systems*, 2019.

[63] Z. Ming, J. Chen, L. Cui, S. Yang, Y. Pan, W. Xiao, and L. Zhou, "Edge-based video surveillance with graph-assisted reinforcement learning in smart construction," *IEEE Internet of Things Journal*, 2021.

[64] C. Yu, J. Liu, S. Nemati, and G. Yin, "Reinforcement learning in healthcare: A survey," *ACM Computing Surveys*, vol. 55, no. 1, pp. 1–36, 2021.

[65] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine uav target tracking with deep reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1522–1530, 2019.

[66] Z. Liu, Q. Liu, L. Tang, K. Jin, H. Wang, M. Liu, and H. Wang, "Visuomotor reinforcement learning for multirobot cooperative navigation," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2021.

[67] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 595–605.

[68] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *CoRR*, vol. abs/1909.12271, 2019.

[69] A. R. Mahmood, D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra, "Benchmarking reinforcement learning algorithms on real-world robots," in *Conference on Robot Learning*, 2018, pp. 561–591.

[70] Z. Hou, Z. Li, C. Hsu, K. Zhang, and J. Xu, "Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly," *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2020.

[71] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3389–3396.

[72] V. Talpaert., I. Sobh., B. Kiran., P. Mannion., S. Yogamani., A. El-Sallab., and P. Perez., "Exploring applications of deep reinforcement learning for real-world autonomous driving systems," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, pp. 564–572.

[73] R. Bellman, "A markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[74] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[75] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, 2000, pp. 1008–1014.

[76] G. Bhatia. Prefabrication in india. [Online]. Available: https://www.commercialdesignindia.com/people/3799-prefabrication-in-india

[77] K. Anastasiades, J. Goffin, M. Rinke, M. Buyle, A. Audenaert, and J. Blom, "Standardisation: An essential enabler for the circular reuse of construction components: A trajectory for a cleaner european construction industry," *Journal of Cleaner Production*, vol. 298, p. 126864, 2021.

[78] D. M. Gann, "Construction as a manufacturing process? similarities and differences between industrialized housing and car production in japan," *Construction Management and Economics*, vol. 14, no. 5, pp. 437–450, 1996.

[79] [Online]. Available: http://www.leoncn.com/news/556.html

[80] P. Cai, Y. Cai, I. Chandrasekaran, and J. Zheng, "Parallel genetic algorithm based automatic path planning for crane lifting in complex environments," *Automation in Construction*, vol. 62, pp. 133–147, 2016.

[81] [Online]. Available: https://www.dwgg88.com/xyzx/wjgjgdbqtj_1.html

[82] S. H. Han, S. Hasan, A. Bouferguène, M. Al-Hussein, and J. Kosa, "Utilization of 3d visualization of mobile crane operations for modular construction on-site assembly," *Journal of Management in Engineering*, vol. 31, no. 5, p. 04014080, 2015.

[83] J. Marshall. London office construction hits record high but covid-19 sends developers' confidence tumbling. [Online]. Available: https://www.building.co.uk/news

[84] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[85] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *The AAAI Conference of Artificial Intelligence*, vol. 30, no. 1, 2016.

[86] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

Aiyu Zhu currently is a Doctoral Candidate in the ISBE group at Eindhoven University of Technology (Tu/e). He received the M.S degree in civil engineering from California State University, Fullerton, US, in 2017. He worked as a BIM engineer at ARCHITECTS and ENGINEERS CO.,LTD.OF SOUTHEAST UNIVERSITY from 2017 to 2019.

His work and interests include robotic construction, AI for construction, Building Information Modelling (BIM), and Linked Building Data (LBD).

Tianhong Dai is a Lecturer in the Department of Computing Science at the University of Aberdeen. He received a Ph.D. in Machine Learning from Imperial College London in 2022, an M.Sc. in Communication and Signal Processing from Imperial College London in 2016, and a B.Eng. (Hons.) in Electronic and Communication Engineering from the University of Liverpool in 2015. He has been a Research Intern in Tencent AI Lab and Huawei Noah's Ark Lab. His research interests include deep reinforcement learning and computer vision.

Gangyan Xu (Member, IEEE) received the B.S. degree in automation and the M.E. degree in systems engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2012, respectively, and the Ph.D. degree in systems engineering from the University of Hong Kong, Hong Kong, in 2016.

He is currently an Assistant Professor with Hong Kong Polytechnic University, Hong Kong. Prior to that, he was an Assistant Professor with the Harbin Institute of Technology, Shenzhen, China, a Research Fellow with Nanyang Technological University, Singapore, and a Research Assistant with the City University of Hong Kong, Hong Kong. His research interests include data-driven optimization and control, intelligent transportation system, resilience engineering, and emergency management.

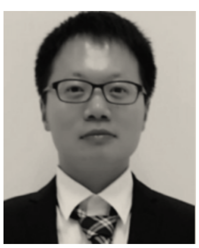
Dr. Xu is an Editorial Board Member of Advanced Engineering Information.

Pieter Pauwels works as an Associate Professor at the Eindhoven University of Technology, the Department of the Built Environment. He previously worked at the Dept. of Architecture and Urban Planning at Ghent University (2008-2019). His PhD research position was a fully funded research position (2008-2012), after which he took on a 100% research oriented Postdoc position at the Institute for Logic, Language and Computation (ILLC) at the University of Amsterdam (UvA; 2012-2014). After 2014, Pieter has been mainly involved in the architectural design and construction industry, aiming to bring building data to the web (linked data, semantic web).

His work and interests are in information system support for the building life-cycle (architectural design, construction, building operation). With a lot of experience and knowledge in computer science and software development, he is involved in a number of industry-oriented research projects on topics affiliated to AI in construction, design thinking, Building Information Modelling (BIM), Linked Building Data (LBD), Linked Data in Architecture and Construction (LDAC), and Semantic Web technologies.

Bauke de Vries is chairing the Information Systems in the Built Environment (ISBE) group, and he is full Professor in the Department of the Built Environment at Eindhoven University of Technology. His main areas of expertise are: Systems Engineering, Building Information Modelling, City Information Modelling, and Process modelling in the context of Smart Cities development.

Bauke de Vries is author of over 100 publications that have been presented on international conferences and scientific articles that have been published in international journals. He is vice-president of CAADFutures; editiorial board member of ITCon, CUMINCAD, IJEPR; conference reviewer of DCC, eCAADE, SIGraDi, CAADRIA, ACADIA, CIB-W78, DDSS, ISARC, ISG; journal reviewer of Computers, Environment and Urban Systems, Journal of Engineering Research, International Journal of E-Planning Research, Journal of Information technology in Construction, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Journal of Building Performance Simulation, Journal of Building Performance Simulation, Journal of Corporate Real Estate, Building Research and Information.



Meng Fang received the Ph.D. degree from the University of Technology, Sydney, Australia. He was a research fellow with the School of Computing and Information Systems, University of Melbourne. He is currently an assistant professor with the Department of Computer Science, University of Liverpool, the United Kingdom. His research focuses on natural language processing and reinforcement learning.