Hybridization of Reinforcement Learning and Agent-Based Modeling to Optimize Construction Planning and Scheduling

Nebiyu Siraj Kedir [a], Sahand Somi [b], Aminah Robinson Fayek [c], Ph.D., P.Eng., and Phuong Nguyen [d], Ph.D.

[a] Hole School of Construction Engineering, Department of Civil and Environmental Engineering, University of Alberta, 7-203 Donadeo Innovation Centre for Engineering, 9211 116 St NW, Edmonton AB, T6G 1H9, Canada; email: nebiyu@ualberta.ca

[b] Hole School of Construction Engineering, Department of Civil and Environmental Engineering, University of Alberta, 7-203 Donadeo Innovation Centre for Engineering, 9211 116 St NW, Edmonton AB, T6G 1H9, Canada; email: ssomi@ualberta.ca

[c] Hole School of Construction Engineering, Department of Civil and Environmental Engineering, University of Alberta, 7-203 Donadeo Innovation Centre for Engineering, 9211 116 St NW, Edmonton AB, T6G 1H9, Canada; email: aminah.robinson@ualberta.ca

[d] Hole School of Construction Engineering, Department of Civil and Environmental Engineering, University of Alberta, 7-203 Donadeo Innovation Centre for Engineering, 9211 116 St NW, Edmonton AB, T6G 1H9, Canada; email: phnguye1@ualberta.ca

Abstract

Decision-making in construction planning and scheduling is complex because of budget and resource constraints, uncertainty, and the dynamic nature of construction environments. A knowledge gap in the construction literature exists regarding decision-making frameworks with the ability to learn and propose an optimal set of solutions for construction scheduling problems, such as activity sequencing and work breakdown structure formulations under uncertainty. The objective of this paper is to propose a hybrid reinforcement learning–graph embedding network model that 1) simulates complex construction planning environments using agent-based modeling and 2) minimizes computational burdens in establishing activity sequences and work breakdown formations. Three case studies with practical construction scheduling problems were used to demonstrate applicability of the developed model. This paper contributes to the body of knowledge by proposing the hybridization of reinforcement learning and simulation approaches to optimize project durations with resource constraints and support construction practitioners in making project planning decision-making.

Keywords: reinforcement learning, agent-based modeling, graph embedding, optimization, planning, decision making

1. Introduction

Construction planning and scheduling is the process of determining what activities are performed and establishing how and when these activities are conducted within the limits of the available time, budget, and resources [1]. According to the Project Management Institute (PMI), planning activities consists of transforming the scope of work to establish a hierarchy of manageable work packages, also called a work breakdown structure (WBS) [2,3], and then determining the sequence of activities' execution according to project constraints including work environment layout, available resources, and scope. In the same manner, construction planning enables a project to accomplish a set of required objectives that can be considered as a two-part problem. First, the solution needs to capture the dynamic construction environment with activities representing project scopes that can be defined as a hierarchy of executable work packages. Second, the solution is a result of estimating duration requirements for activities and optimizing activity sequencing based on multiple and pre-determined constraints that also incorporate decision makers' knowledge and experience. Construction planning includes scheduling and other forms of planning, such as material handling, site layout planning, equipment path planning, and site logistics planning [4]. Scheduling problems are an important part of construction planning activities in terms of planning physical construction project components that have a specified set of start and finish timelines and an estimated duration.

Researchers have proposed multiple decision-aid methods, such as simulation, optimization, multi-criteria decision-making, and automation, to tackle activity sequencing and WBS formations in construction scheduling problems [4]. Some methods include linear programming, heuristic or meta-heuristic approaches, and hybrid simulation approaches such as discrete event simulation-genetic algorithm (DES-GA). These methods have proposed solutions by solving mathematical objective functions that optimize a given metric, such as time, cost, resource, or quality. These approaches have some shortcomings in capturing uncertainty in the construction environment, raising computational burdens, and not being easily generalizable to multiple construction projects. In a scheduling problem, the optimization process needs to consider multiple constraints tied to each activity, such as time, budget, and resources. These constraints can include 1) precedence relationships, 2) project manager preferences, such as activity associated with a rented crane may need to take precedence to minimize equipment rental costs, and 3) interruptions, such as equipment breakdowns. To tackle these constraints, methods are needed that can capitalize on the simulated environment to understand complex behaviors and derive more sufficient decisions.

Reinforcement learning (RL) is very effective for decision-making processes in construction problems. RL algorithms are able to solve optimization problems with higher constraints [5] and perform efficiently with increasing complexity and number of activities [6]. The RL agent learns to implement better actions, including optimal sequencing of activities, through training achieved from exploiting local rewards and exploring random actions despite lower rewards. Hence, RL can help fill the aforementioned shortcomings of current decision-aid methods in construction planning by developing a local decision-making policy for each agent, based on communication channels, and by breaking down the problem into sub-problems, all of which contributes to computational efficiency. Using RL assists construction practitioners in facilitating generalizations through the learning process, because different problems can be broken down into similar sub-problems. Moreover, RL facilitates agent communications and enables agents to arrive at a set of decisions involving a set of joint actions. This results in a faster convergence to the optimum global policy. However, an RL process does not capture the dynamic nature of modeling in the construction environment, because of the complexity caused by various interactions between system components [7]. In a construction setting, however, having a model of the construction environment is crucial.

Simulation techniques have been used to capture the dynamic nature of the construction environment as well as uncertainties in the modeling process [8]. Compared to other simulation techniques, such as DES and system dynamics (SD), agent-based modeling (ABM) is able to handle these complexities and capture emerging behaviors. ABM is capable of handling very complex real-world systems often containing large amounts of autonomous, goal-driven, and adapting agents [9]. ABM uses a bottom-up approach where the system is described as interacting objects with their behaviors, which allow complex emergent behaviors to be captured. ABM enables tracking of agent interactions in their artificial environments to understand overall processes that lead to global patterns [10]. By incorporating ABM in an RL process, necessary features that support environment modelling, such as system parameters, system behaviors, and rules, are provided in order to enable an efficient representation of the dynamic construction environment and provide the RL platform with the necessary features to support environment modelling.

The objective of this paper is to propose an RL-ABM method with graph networks that can be used to support decision-making in construction planning by providing optimum work package sequencing to schedule activities based on project constraints. The application of the proposed model can be extended to establishing a WBS for a construction project. Three case studies were used to demonstrate the proposed model and discuss the applicability of RL-ABM to addressing similar problems related to activity sequencing. The developed RL-ABM method enables construction decision-makers to evaluate project objectives, facilitates the optimization of multiple types of resources during planning through the RL agent's learning ability, is able to incorporate resource planning during schedule development, and can be generalized to other construction planning problems. Moreover, the applications of the method can be extended to scope definition (WBS formulation) at the project level in future work that will extend this study.

The rest of this paper is structured as follows. First, as background, a literature review section is presented, which discusses decision-making in construction planning and shortcomings of current decision-aid approaches to scheduling problems, followed by an introduction of simulation approaches and RL to address the gap in the literature. Next, the theoretical development of RL-ABM is presented as part of the proposed methodology, which also includes the steps of problem definition, ABM simulation, and development of the RL model. Three case studies are then presented to demonstrate application of the proposed RL-ABM method. Finally, conclusions are presented and recommendations for future work are discussed.

2. Background

This section provides an overview of decision making in construction planning. Simulation approaches and RL are then discussed along with the knowledge gap existing in the construction planning literature.

2.1. Decision-making in construction planning

Decision-making is a critical aspect of construction processes such as policymaking, budgeting, risk and safety, planning and scheduling, bidding and tendering, productivity, and performance [11–13]. In construction planning and scheduling, decision-making–related problems consist of determining the optimum sequence of activities according to project objectives and constraints, and then defining the WBS [14]. For various optimization problems, current construction planning approaches mostly comprise one of or a combination of the following: expert opinion and experience, mathematical and heuristic formulations, intelligent methods, evolutionary methods, and simulation techniques. Methods involving expert opinion and experience can exhibit potential uncertainty and might not significantly benefit objective problems that involve rigorous computation [15]. Mathematical methods, such as integer, linear, or dynamic programming, are computationally cumbersome, complex, and easily trapped in a local optimum [16]. Heuristic methods are a collection of proposed rules that do not use rigorous mathematical formulations [17] and offer a much simpler approach using rules-of-thumb and experience [16]. Some examples of heuristic and meta-heuristic approaches can be found in the work of Sonmez et al. [18], Yahya and Saka [19], Liu et al. [20], and Chen and Shahandashti [21]. Heuristic methods perform differently in different problem contexts and do not always guarantee optimum solutions, as no direct approach exists for selecting the best heuristic approach [22]. In situations where insufficient data is available for modeling and computing processes, intelligent methods [23–25] could be used to establish WBS and identify the proper sequence of activities. Evolutionary methods can become difficult to implement and make the computation process extremely intensive and expensive to perform [26]. Some studies [27–29] have also proposed hybrid simulation approaches that simulate construction problems using a simulation approach (such as DES) and an optimization method. This paper presents an alternative to other methods currently found in construction planning literature: a simulation engine that provides a scientific method for finding an optimal set of solutions for particular scheduling problems by simulating the environment, which consists of activity durations, resource availabilities, and precedence relationships, in an optimization platform, which takes into account the objective function and pre-defined constraints.

2.2. Simulation approaches in construction

Simulation as a scientific tool for analyzing complex behaviors and processes in construction projects was first introduced in the 1960s by Teicholz [30] via a "link-node" model to investigate simple networking concepts and explain construction operations. The first software implementation of DES is believed to have been introduced by Gordon [31]. Some examples of DES application in the construction industry include construction planning and project scheduling [32,33], estimation in construction processes [34–36], productivity and performance [37–39], and construction simulation [32,40,41]. Despite the capability of DES to simulate process-type systems, DES elements behave in a predetermined manner ignoring unique operational real-life scenarios that occur as a result of resource constraints. For many construction systems with complex project scenarios, such as earthwork operations including a large number of equipment types, varying arrival, service, breakdown processes, and weakly defined haul-road networks and volumes, more entities are required to account for the increasing complexity, making DES approaches computationally demanding. Zankoul et al. [42] compared DES with ABM for the same earthmoving project and showed that DES had increased computational burden due to additional entities needed to represent the system.

Agent-based modeling (ABM) surpasses earlier methods such as DES, as it can be used to capture emerging behaviors that result from complex interactions of interrelating model components [7]. ABM is a computer simulation technique that enables prediction of overall system behavior and emerging patterns by modeling the behaviors of system components as well as individual agents [43]. Agents are discrete entities whose descriptors can be a type, such as "construction worker" that have their own attributes, such as "age," "workstation," "assigned task," and "behavior." ABM can be used to model interactions of individual agents with each other and with their environment [44]. Examples of ABM applications include scheduling and planning [45–47] and decision making [11,48]. ABM is an appropriate tool for describing complex systems with dynamic processes of agent interactions that are repeatedly simulated over time [49], because competitive and repetitive interactions between agents can result in extremely complex behaviors [50]. In this regard, ABM can easily handle a large number of activities with differing attributes and allow for a better representation of complex relationships between those activities, such as precedence relationships, competitions for resources, and changing construction conditions, which makes this method ideal to simulate construction environments for planning and scheduling purposes.

2.3. Reinforcement learning (RL)

RL settings can be classified as single-agent RL or multi-agent RL (MARL) depending on the number of autonomous agents that influence the system's state and reward [51]. RL can also be classified as model-based or model-free RL [52]. In terms of its applications, RL has been used in various applications in the field of civil engineering owing to its capabilities that make it particularly successful in solving complex problems [53]. Some of these applications include works in the area of design and operations for water structures [54,55], transportation engineering [56–58], and maintenance [59]. RL has been effectively applied to develop strategic conventional tunneling in construction, which provided optimal economic and safe policies with potential to discover new tunneling strategies [60]. RL is also emerging as a control technique [61], and it is of growing interest in research, with demonstrated potential particularly in enhancing building performance [62–65]. Because RL uses an intelligent agent to learn to make a series of optimal decisions [52], it is a suitable approach for performing construction planning where a series of decisions (e.g., activity sequencing, resource allocation) are performed at different times throughout a project's lifecycle. In the area of scheduling, the majority of RL-based research has focused on production scheduling. Creighton and Nahavandi [66] proposed an intelligent agent-based scheduling system that uses DES as a simulation engine with the goal of minimizing total production costs depending on job sequence and batch size. Cao et al. [67] proposed an RL model using Monte Carlo simulation to solve a production planning problem that minimizes inventory and penalty costs. Wei and Zhao [68] used Q-learning algorithm to schedule a dynamic job-shop problem that considers machine selection. Zhang et al. [69] used an RL method coupled with heuristic method and simulation to perform parallel machine scheduling that minimizes mean flow time of jobs. Fonseca-Reyna et al. [70] used RL to solve a scheduling problem that finds a permutation of operations that is processed sequentially on a set of machines with the objective of minimizing the completion time of all jobs. Bouazza et al. [71] used an RL approach with Q-learning to solve a job-shop scheduling problem.

Unlike supervised and unsupervised learning approaches, RL is a machine learning technique that uses the environment for learning and is not dependent on a predefined dataset [72]. Moreover, RL is particularly advantageous in the area of sequential decision making, which is a key challenge in artificial intelligence research [73]. When sequential decision making is formalized as Markov decision process (MDP) framework optimization problem, selecting the sequence of actions that produce optimal results (e.g., path planning) becomes complicated because of inherent key elements of the world (i.e., information about the environment and states; influence of actions on the environment; the notion of preferred actions now and in the future) [73]. In this regard, RL can offer an efficient solution for construction operation problems that may be viewed as a collection of recurring activities [53] where the objective is to produce an optimal solution (i.e., optimal project performance measure such as minimum project duration or minimum cost) in a dynamic environment (i.e., changing project conditions) subject to constraints (i.e., limited resources). RL's capability also extends to solving large-scale dynamic optimization problems and complex multi-objective sequential decision-making problems [73].

Even though there is growing research into RL-based optimization approaches that demonstrate the benefits of RL method in other fields within construction, most applications of RL for scheduling problems with respect to improving production have been limited to the manufacturing sector. In construction planning, decision makers analyze various activities to ensure optimal use of available resources and achieve required performance to meet project objectives with respect to cost, time, and quality. Establishing WBS and activity sequencing requires consideration of numerous interacting factors between the activities themselves, such as technology constraints, precedence relationships, available resources, conflicting objectives, and incomplete information. In this regard, RL enables a model to process optimization approaches that provide human-like intuitions and learning capabilities, which can enable decision makers to obtain better solutions that can adapt to changing environments.

2.3.1. Markov decision process (MDP)

Markov decision process (MDP) is a framework that describes the process of learning from interaction with the environment in order to achieve a goal. MDP has five components [74]: 1) the set of possible actions ($A_t \in$ A) that can be taken by the agent or the decision-maker; 2) the set of all possible states ($S_t \in$ S) that can be experienced by the agent; 3) the immediate reward r that is received by the agent corresponding to the given state and action pair, defined in Eq. (1); 4) the discount factor γ that signifies the relative importance future rewards have compared to the current immediate reward, defined in Eq. (2), which denotes the discounted cumulative reward G_t following time *t*; and 5) the transition probability $p(s',\ r \mid s,\ a)$ of a state corresponding to past state and action, defined in Eq. (3). The agent-environment interaction in MDP is summarized in Fig. 1.

$$R(s,a) = E[r_{t+1} \mid S_t = s, A_t = a] \quad (1)$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad (2)$$

$$p(s',\ r \mid s,\ a) \doteq \mathrm{p}(S_t = s',\ R_t = r \mid S_{t-1} = s,\ A_{t-1} = a) \quad (3)$$

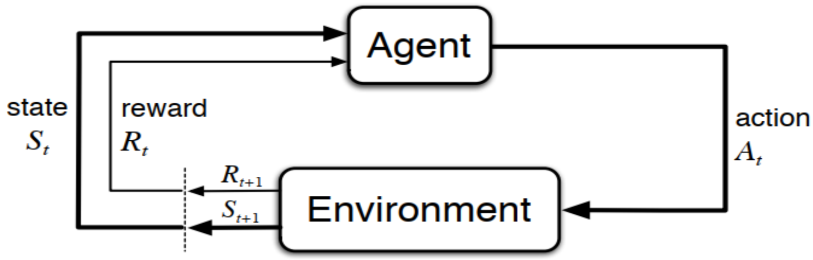

Fig. 1. Agent-environment interaction in MDP (adapted from [52]).

In MDP, the optimal policy $\pi^*(a|s)$ can be the function that maps the current state s to the best action a^* while maximizing the expected future reward, as shown in Eq. (4).

$$\pi^* = argmax\ \mathbb{E}[G_t|S_t = s, A_t = a] \quad (4)$$

2.3.2. RL algorithms

RL algorithms for solving an MDP problem can be implemented in two ways: through 1) action-value approximation or 2) policy approximation. Action-value methods directly learn the expected return of taking each action a in a specific state s [52]. The action-value function $q_\pi(s, a)$ is defined in Eq. (5), and the optimal action-value function for the optimal policy (π^*) is defined in Eq. (6) by considering the Bellman optimality equation, Eq. (5), and Eq. (3):

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] \quad (5)$$

$$q_{\pi^*}(s, a) = \sum_{s',r} p(s', r|s, a)\,[r + \gamma \max_{a'} q_{\pi^*}(s', a')] \quad (6)$$

On the other hand, in some MDPs, directly learning action-value functions is challenging in a big action space, and as a result, the policy function is used to calculate the preferences for each action in each state. The parameterized policy formula is defined in Eq. (7).

$$\pi(a|s, \theta) = Pr[A_t = a|\ S_t = s, \theta_t = \theta] \quad (7)$$

Eq. (7) presents the probability of selecting an action as action preference. For example, this probability could be a linear function of any complex structure of deep learning, where θ is the weights or parameters of the function. Eq. (8) and Eq. (9) express the discrete action space for a linear parameterized policy with soft-max distribution [75]. The objective in RL processes is to learn q^* or θ^* by interacting with the environment and receiving rewards. This learning is accomplished by updating a policy or set of action-value function parameters, which means learning the best values for each state or sub-problem, which leads to solving the MDP.

$$\pi(a|s,\theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}} \qquad (8)$$

where

$$h(s,a,\theta) = \theta^T x(s,a) \qquad (9)$$

3. Methodology

The research methodology of this study consists of four steps: 1) development of the RL model, 2) problem definition, 3) ABM simulation process, and 4) development of the RL process for construction planning.

3.1 Development of RL model

3.1.1. MDP states and actions

In the construction environment, formalizing resource-constraint scheduling as an MDP is described as follows. Possible actions ($A_t \in$ A) are activities that can be scheduled according to project state ($S_t \in$ S). Project state in this study is characterized by project time, available resources, and the state of each activity in the network. Each activity has four states, namely "NotReady," "Ready," "InProgress," and "Complete," and each state is represented in a binary format. Hence, the MDP environment for the scheduling problem starts by defining which activities can be used to prioritize schedules and thus minimize project total finish time *T*. At each step, the environment advances to the nearest finish time of activities in order to update the background, and then, based on project state, possible activities are scheduled from the pool of possible actions. In the scheduling problem, the reward is considered as a negative value of time, and the objective is to maximize the long-term reward. In this sense, maximizing over negative value results in minimizing the total project time. The "state" and "action" pairs, which are the two major components in the MDP, are described below.

State: The construction scheduling problem is formulated as an MDP problem with RL algorithms that use an MDP framework to derive optimal strategies. Each state in the RL algorithms is represented in a structure format as an input to calculate future values according to possible actions in the current state. For the scheduling problem regarding resource constraints, each state corresponds to the activity on node (AON) network at a given timestep. Therefore, each state S represents the outcome of a previous action and comprises the following information:

i. Activities states in simulation: These can be obtained from the simulation model at each timestep per a corresponding numeric value, as shown in Table 1.

Table 1: Description of activities.

State	State description
0	NotReady
1	Ready
2	InProgress
3	Complete

ii. Available resources: The current availability of resources should be present in the state information, because they are required to assess which actions can be performed next.

iii. Activities duration: The state gives information on the activities' duration.

Action: For each state, the agent selects an action from the available activities, which affects the resource pool of activities. Hence, selecting an action results in changing the project state, and the agents use updated information to select the next action. In other words, agents select one action per state.

3.1.2. In construction planning environments, the agents select an environment action ($A_t \in \mathrm{A}$) that affects project total duration. These agents learn to make the optimal sequence of decisions that can meet the predefined objective by maximizing the received reward for a given action while also exploring the decision space to avoid local solutions, as shown in Eq. (10).

$$q_{\pi}(s,a) = \mathbb{E}_{\pi}[G_t|S_t = s, A_t = a] q_{\pi^*}(s,a) = \sum_{s',r} p(s',r|s,a)\,[r + \gamma \max_{a'} q_{\pi^*}(s',a') \pi(a|s,\theta) = Pr[A_t = a|\, S_t = s, \theta_t = \theta] \pi(a|s,\theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}\ h(s,a,\theta) = \theta^T x(s,a) \qquad (10)$$

The value function therefore learns to calculate the value of each possible activity based on receiving rewards and tries to estimate the priority of the activities according to the project state. Fig. 2 provides an example of how the RL agent performs the optimization process to produce an improved network diagram. In Step 1, the RL agent observes the current state of the AON to recognize the resource requirements, initial project network with technology constraints, and the duration of each activity. In Step 2, the agent prioritizes what action to take based on the current state and reward system of the RL algorithm. In Step 3, it takes the action to start activity A, based on priority rules and agent preferences from the previous step. In Step 4, the RL agent observes the next state and updates the AON network based on the previous action taken. As a result, the path from A to B is resource constrained in order to minimize total project duration from 3 days to 2 days.

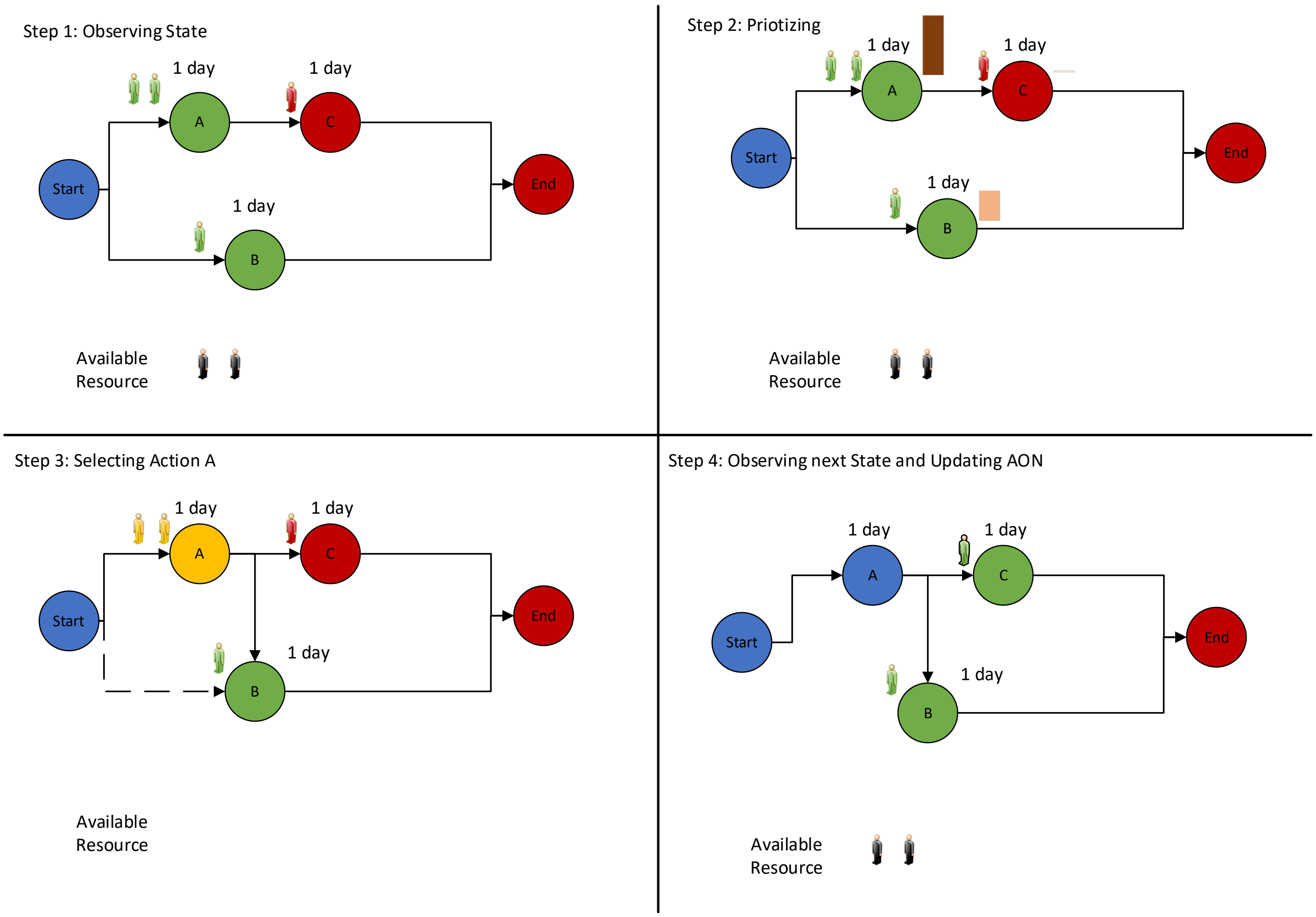

Fig. 2. RL process for optimizing AON.

3.2. Problem definition

Construction practitioners can be faced with several combinations of planning issues related to factors such as time, cost, and quality. Additionally, construction projects are usually executed under resource constraints related to labor, material, and equipment. Therefore, the planning process aims to optimize the use of resources and to sequence activities in order to meet project objectives. The problem in this study is defined as scheduling the network of construction activities that are subject to resource constraints with the objective of minimizing the total project duration.

Each construction activity in a given project has its own normal activity duration, signified by the amount of time required to complete such an activity under normal circumstances. The duration for the assigned activities is measured in increments of time called *planning units*. In this study, these activities are sequenced to comply with project schedule requirements in order to complete the overall project with the shortest possible duration.

Eqs. (10–12) show the logic for resource allocation optimization:

$$\textit{minimize} \qquad T = \max\{t_i + d_i \mid i = 1, 2, \dots, n\} \qquad (10)$$

$$\textit{subject to} \qquad t_j - t_i - d_i \geq 0 \quad j \in S_i \qquad (11)$$

$$\sum_{t_j \in A_{t_i}} r_{d_{ik}} \leq b_k \quad (k = 1,2, \dots, m) \qquad (12)$$

where T = project duration; $t_{i,j}$ = starting date of activity *i, j*; d_i = duration of activity *i*; A_{t_i} = set of ongoing activities at date t_i ; and b_k = resource limit of k^{th} resource.

Eq. (10) indicates the computation for project duration. Eq. (11) indicates that the difference between the occurrence times of two connected nodes should be greater than or equal to the duration of the connecting activity. Eq. (12) imposes the restriction on utilization of resources, which can not exceed available resources. The proposed model for solving this scheduling problem is shown in Fig. 3 and elaborated in the subsequent sections. The proposed model starts with the ABM component, where the construction activities are analyzed using critical path method (CPM) and then used to create the model environment. The RL component consists of establishing the graph embedding network in an ABM environment to optimize the duration of the scheduling problem.

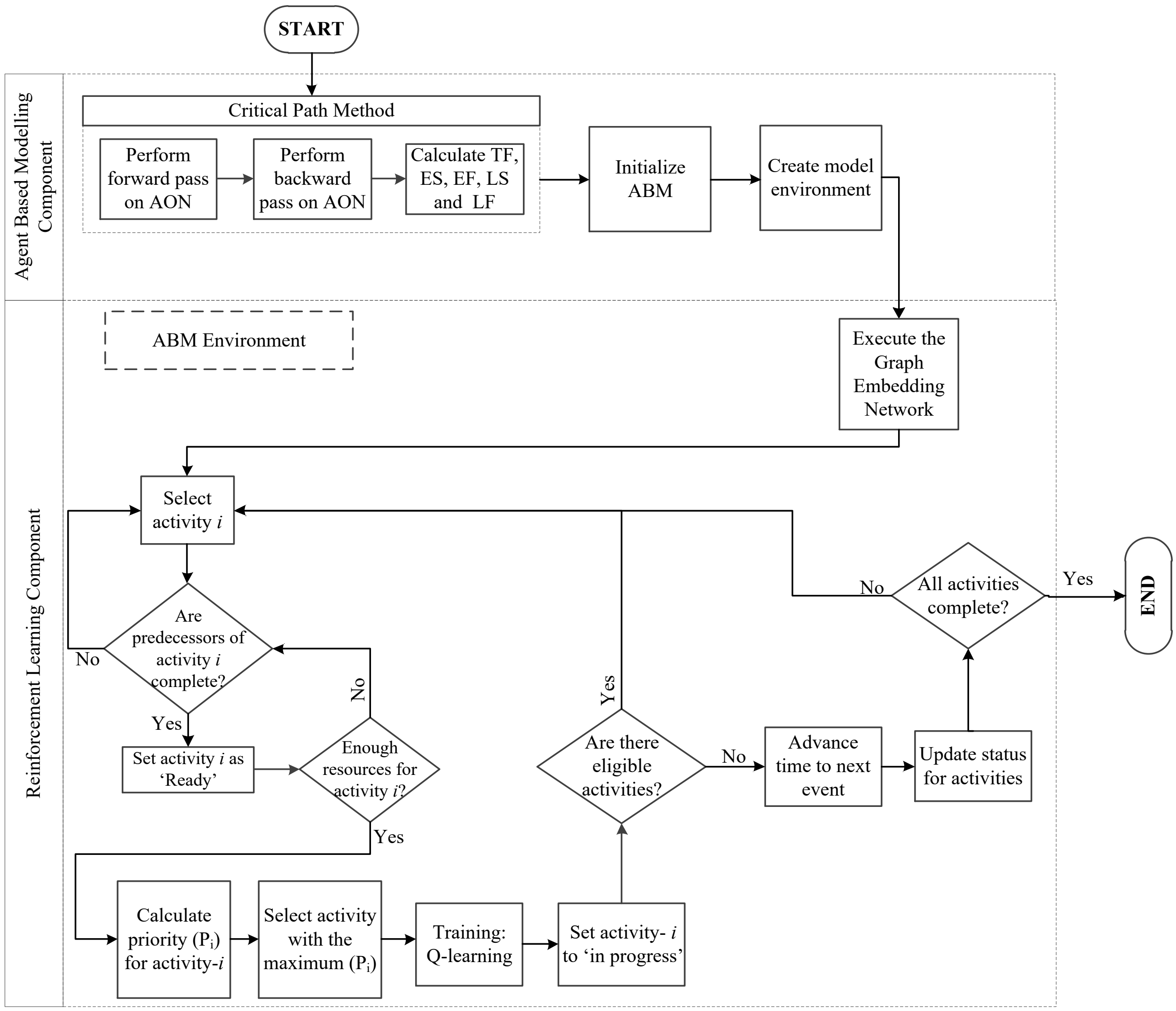

Fig. 3. Flow diagram of the proposed RL-ABM method.

3.3. ABM simulation

The optimization problem introduced in Eq. (10) is solved using the RL approach. In this section, ABM simulation is discussed in order to define the environment for the RL optimization platform. The ABM is used to define the environment, which consists of the intelligent RL agent and the activity agents representing activities of the project.

3.3.1. Input to ABM simulation

The input to ABM simulation was the characteristics of the AON network, which holds the project information related to the sequence of activities that comply with technological constraints. These activities are connected in a finish-to-start manner, where the end of the preceding activity marks the possible beginning of subsequent activities. This information was used to define the project environment in the ABM platform. The main advantage of using ABM is to enable the creation of the RL environment, which can be used by the RL agent to obtain the current state of the system and facilitate the optimization process.

3.3.2. ABM simulation process

Using the given AON network, the early start (ES), early finish (EF), late start (LS), late finish (LF), and total float (TF) of each activity agent is calculated using CPM [76]. These values are used as RL agent parameters and processed by the intelligent RL agent. The activity agent is the main agent in the proposed ABM and the main driver of the simulation. Activity agents could be considered as goal-oriented reactive agents whose sole purpose is to be completed. An activity agent transitions into different state-charts by starting, performing certain tasks for a given duration, then concluding. In addition to the information on states, the activity agent includes the list of resources and predecessors for each activity and the normal duration associated with it.

Fig. 4 shows the states of an activity agent considered in the ABM simulation. All activities start in a “NotReady” state, which signifies the initial state of all activities and the states of all other activities whose predecessor activities have not been completed. Next, each activity checks if its corresponding predecessors are completed. This check is completed by making sure the conditional statements are returned as 'TRUE' for initial technological constraints within in each activity agent. After confirming this check, the activity transforms to a "Ready" state. An activity in "Ready" state then checks if its corresponding resources are available to start the activity and move to the "InProgress" state. Multiple activities in the "Ready" state will compete for similar resources based on present priority rules. The RL agent checks whether enough resources are available for an activity in the "Ready" state. In this stage, if a predefined priority rule exists (e.g., activities with longer duration get preference; activities with lesser number of resources get preference), the activity agents utilize that priority rule to capture the required resources and transition to the next state. Otherwise, the agent assigns priorities for potential activities based on its deep neural network and selects the highest-priority one to proceed. An activity remains in a “Ready” state if there are not enough resources available.

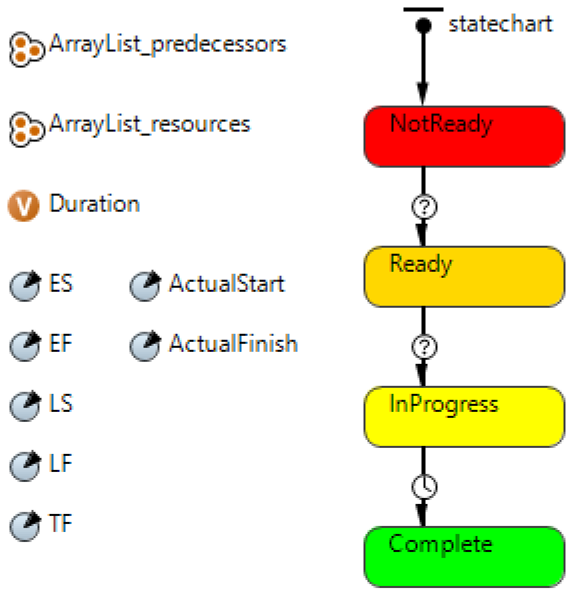

Fig. 4. States of an activity agent considered in the ABM simulation (where ES is *early start*, EF is *early finish*, LS is *late start*, LF is *late finish*, and TF is *total float*).

3.4. Implementation of RL model in construction planning problems

3.4.1. Input to RL modelling

As noted in section 3.3.2, ES, EF, LS, LF, and TF are the parameters calculated using CPM. As part of the defining features of each activity, these parameters are used as inputs for the RL agent's deep neural network. Agent-based modelling output is used as input to define states and actions for each step of optimization.

3.4.2. Graph embedding network

As noted in section 2.3.1, agents select one action per state in the MDP. To model an action, a graph neural network structure is used to address the challenges of project size and modeling relationships between activities. One of the biggest challenges for this type of optimization is the running time corresponding to the number of activities or actions, which depends on project size. Graph neural network applications in similar problems show great performance because instead of a complex network, the whole graph consists of a simple neural network mostly with one hidden layer, which decreases the required computational resources needed for calculation. Regarding modeling relationships between activities, it is very important to understand precedent relationships between activities, which can significantly impact scheduling. In the graph neural network architecture, this important feature can be easily modeled and used to help RL agents to make optimal actions.

With the objective to optimize duration over a project network, graph *G* is defined according to the project network, in which nodes represent activities and edges are used to represent pre-defined technological constraints. After defining graph *G*, the graph structure is converted to vectors to represent such complex phenomena. In this study, a deep learning architecture is leveraged over the graph, in particular *structure2vec* [75]. In this study, the value function was the result of *structure2vec of environment* according to project state.

3.4.3 Parameterizing Q-function

Parameterization of Q-function is performed using the embeddings from *structure2vec*. Eq. (13) [75] shows the design of *F* to update a *p*-dimensional embedding μ_v^t as:

$$\mu_v^{t+1} \leftarrow relu\left(\theta_1 x_v + \theta_2 \sum_{u \in N(v)} \mu_u^{(t)} + \theta_3 \sum_{u \in N(v)} relu\left(\theta_4(v,u)\right)\right) \tag{13}$$

where x_v is a binary scalar of activity state; *relu* is the rectified linear unit ($relu(z) = max(0,z)$) applied elementwise to its input; and $\theta_1, \theta_4 \in \mathbb{R}^p$ and $\theta_2, \theta_3 \in \mathbb{R}^{p \times p}$ are the model parameters.

Next, Q-function is defined as shown in Eq. (14):

$$Q = \theta_5^T \, relu\left(\left[\theta_6 \sum_{u \in V} \mu_u^{(T)}, \theta_7 \mu_v^{(T)}\right]\right) \tag{14}$$

where $\theta_5 \in \mathbb{R}^{2p}$ and $\theta_6, \theta_7 \in \mathbb{R}^{p \times p}$.

Q-function depends on a collection of seven parameters. For the graph embedding computation, the number of iterations *T* for the graph embedding computation is typically small (i.e., T=4) [75].

3.4.4. Training: Q-learning

Two distinctions are made, where the term “episode” refers to the complete sequence of activities from simulation start to termination. A single step within an episode is one action, such as an “InProgress” activity. In this regard, the Q-learning performs a gradient step to minimize the squared loss, as shown in Eq. (15), by updating the function approximator's parameters:

$$(y - Q(v))^2 \tag{15}$$

where $y = \gamma \max_{v'} Q(v') + r(S_t, v_t,)$ for a non-terminal state S_t.

3.4.5. RL-ABM simulation

To optimize project duration while allocating labor resource appropriately, the proposed RL-ABM method consists of three phases. First, the ABM platform performs forward and backward passes to the AON network to obtain the initializing parameters described in the ABM model. The AON diagram then serves as the environment of the RL model, where initial sequencing requirements are fulfilled according to technology constraints. Each activity is given an initial "Not-Ready" state. Second, the AON is transformed into a graph network so the RL agent understands the position of each activity in the overall AON network. In this step, additional identifiers of each activity (i.e., duration of task, required resources, dependency relationships) are used as inputs to form the graph network using the Networkx library. The architecture of the graph neural network for an example AON of five activities (i.e., A, B, C, D, E) is shown in Fig. 5. In this regard, each activity is defined by eight attributes to be used in the RL platform. The first four attributes represent the quaternary value of an activity's state as defined in Table 1. The remaining four attributes capture the resources available (r), duration of the activity (d), and two attributes for the position of the activity in the AON (edges). Third, the RL platform executes the RL optimization algorithm, which uses Q-learning to select an action, calculate action values, and learn to perform activity sequencing that satisfies resource requirements and minimizes project durations. In this regard, the graph neural network class is defined with the PyTorch library. The graph neural network has eight layers to compute the value function. The first layer computes the nodes' values based on defining attributes, while the second layer computes the values of neighbor nodes. The third and fourth layers compute edged values. The last four layers convert the value of each node to a vector value. In the graph neural network, to reduce loss in calculating neighbor values, the values are calculated in three iterations. In the RL section, the code utilizes Q-learning with ε-greedy policy. The reply buffer class saves experience for re-calculating values in order to train the network weights for improving the learning value network.

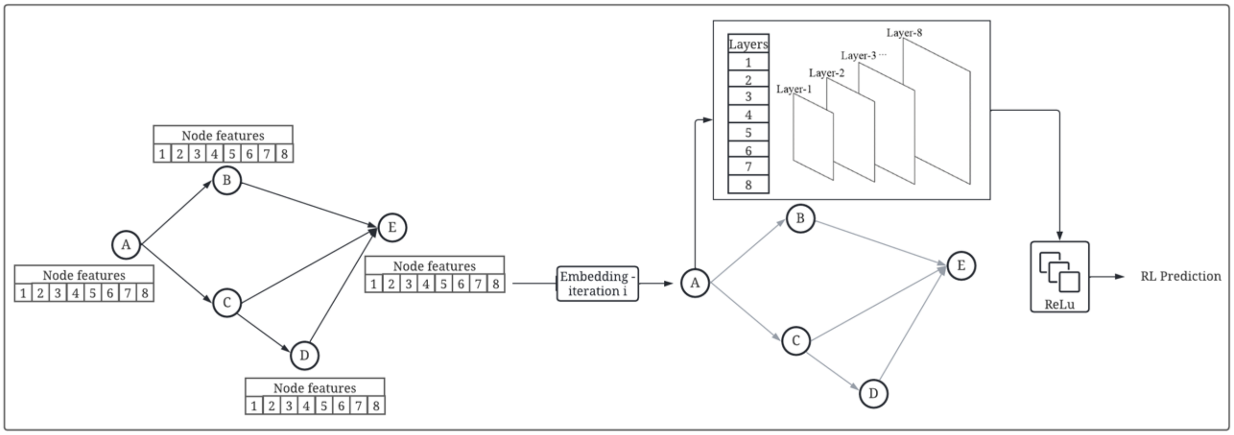

Fig. 5. Graph neural network for an example AON of five activities.

3.4.6. Output from RL-ABM simulation

The simulation outputs integrate processes executed in the RL and ABM. The ABM simulates the resource-constrained activity scheduling to produce the outputs resulting from the optimization performed by the RL platform. The resulting output consists of a modified project AON network that sequences 1) the set of activities to satisfy resource constraint requirements and 2) the activities to optimize with pre-determined objectives, such as duration and cost. This guides construction practitioners in performing the set of activities in an optimal and informed manner, executing the planning process efficiently, and meeting project objectives. The proposed RL model addresses the uncertainties that arise from assigning durations for activities. In the overall process of construction scheduling, a probabilistic approach is used that assigns a triangular distribution of duration for each activity. The consequent uncertainty in the overall scheduling problem, resulting from the dependencies and relationships between individual activities, is solved in the RL platform via coding that accounts for such types of uncertainty. Compared to other scheduling optimization methods, the RL agent can be modeled to arrive to a policy that finds the shortest possible project duration. However, for this study, the uncertainties stemming from activity duration assignments are assigned in a deterministic manner to provide straightforward comparisons to the case studies referred from Lu and Li [68].

4. Case studies

To demonstrate the proposed RL-ABM methodology, this study utilized construction planning case studies elaborated from three scheduling problems. The first two are described in Lu and Li [77]. Case study 1 illustrates how to utilize the proposed RL-ABM method to address a simple scheduling problem. Case study 2 demonstrates the applicability of the proposed model in construction planning to address a more complicated scheduling problem from a bridge construction project. Case study 3 is a more complicated scheduling problem adapted from Zhang et al. [78], selected in order to further demonstrate the methodology.

4.1. Case study 1

The first case study included a simple network with nine activities with the one resource type of labor, for a simple scheduling problem [77]. The resource in this case study was limited to four units of labor per day. The AON network is illustrated in Fig. 6, and the structure of the activity table is shown in Table 2.

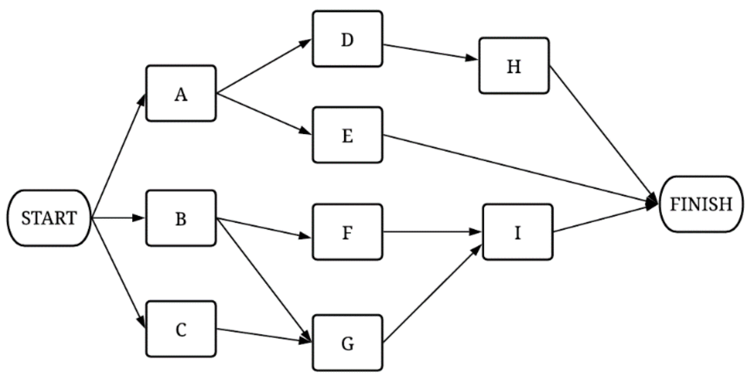

Fig. 6. AON network of case-study 1, a simple scheduling problem (adapted from [77]).

Table 2. Structure of activity table for case study 1, a simple scheduling problem (adapted from [77]).

Node number	Activity	Duration	Resource	Predecessor(s)
1	A	2	4L	-
2	B	3	4L	-
3	C	5	4L	-
4	D	4	3L	A
5	E	4	1L	A
6	F	3	2L	B

7	G	6	2L	B, C
8	H	2	2L	D
9	I	3	2L	F, G

In case study 1, the learning rate and ε value were scheduled to decrease during learning. As a result, in the first episodes of learning, the code attempted to use more random actions. However, in the middle of training, since the ε value was less than 0.5, the network dominated the RL decision-making process. In this problem, since this case study project had only one type of resource, the RL agent found the best policy by prioritizing the sequence of activities based on their TF. For training the model, some hyperparameters need to be set in order to achieve optimum performance of training. Hence, for the learning process, four important hyperparameters directly affect the speed of convergence to the optimum policy: number of episodes, memory capacity, number of steps to update GNN, and batch size. In this case study, these values were set to 4000, 10,000, 2, and 16, respectively.

The result of the AON network using the proposed RL-ABM method for case study 1 is shown in Fig. 7, and the corresponding Gantt chart is shown in Fig. 8. In this regard, the result of the RL-ABM algorithm improved the result of the total duration of this network by a total of 3 days compared to the previous research of Lu and Li [77].

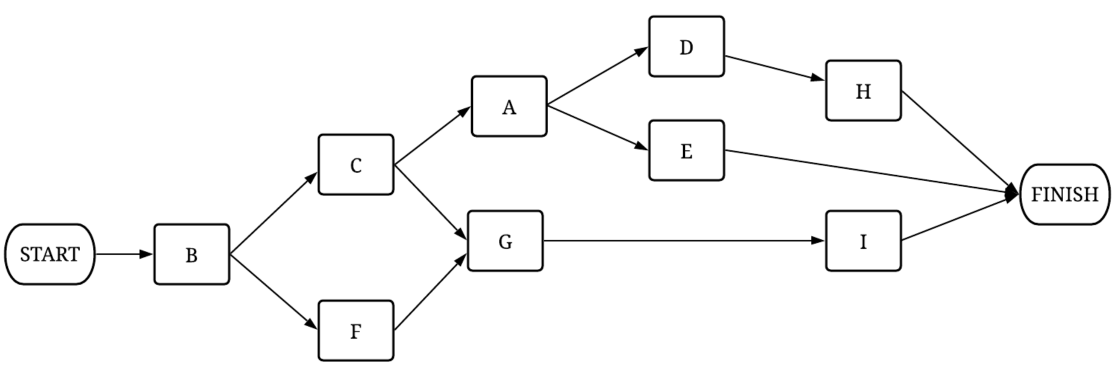

Fig. 7. Resulting AON network for case study 1 based on the proposed RL-ABM method.

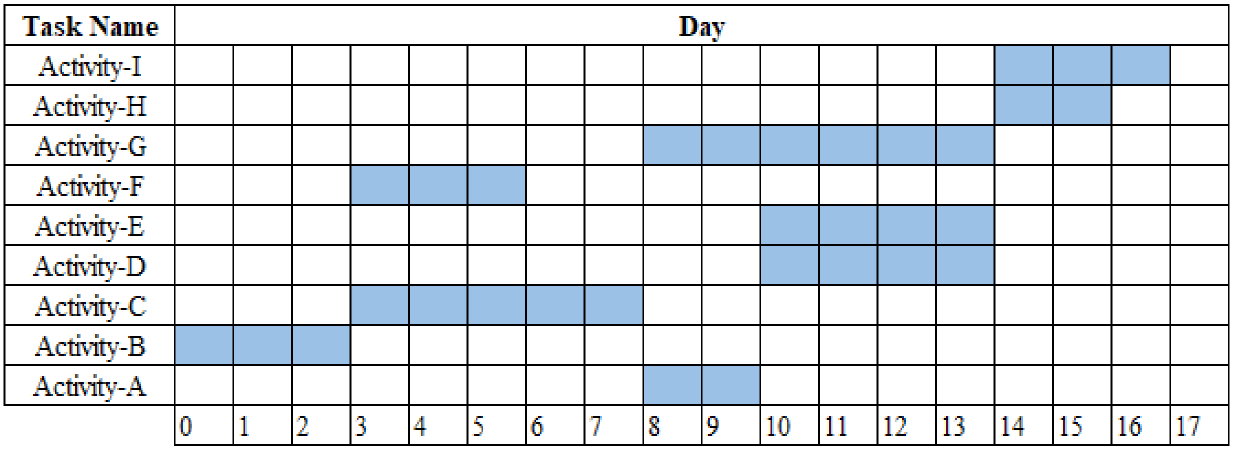

Fig. 8. Corresponding Gantt chart for the RL-ABM solution for case study 1.

4.2. Case study 2

The second case study is a scheduling problem in a bridge construction project [77]. A sample application of the proposed RL-ABM method was based on a local project of constructing a small footbridge, consisting of three stages of construction and requiring multiple types of resources. Two abutments, including footers and supports, were constructed in Stages 1 and 2, respectively, which were reinforced cast-in-place concrete structures. Stage 3 was erection of the superstructure, which was prefabricated in a remote steel plant and moved to the site for installation. The project network is shown in Fig. 9, and Table 3 lists the duration and resource requirements for each activity. Available resources were six skilled laborers (LB), one set of rented formwork for concreting footer and abutment (FM), one excavator (EX), two mobile cranes (MC), and one set of prefabricated steel superstructure (ST) scheduled to be moved to site on day 17. The work contents of identical activities on two stages were slightly different because of particular site conditions and slight design variations on each abutment. In this bridge project, the labor work content of multitasking skilled laborers was the primary criteria in deciding the priority for assigning resources to competing activities.

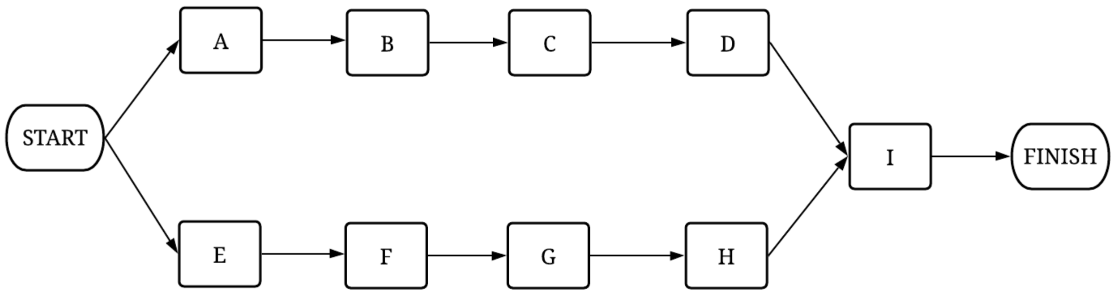

Fig. 9. AON network for case study 2, the bridge construction problem (adapted from [77]).

Table 3. Structure of activity table for case study 2, the bridge construction problem (adapted from [77]).

Node Number	Activity	Description	Duration	Resource(s)	Predecessor(s)
1	A	Excavation stage 1	2	2LB, 1EX	-
2	B	Formwork stage 1	3	4LB, 1FM, 1MC	A
3	C	Concrete stage 1	5	4LB	B
4	D	Backfill stage 1	4	2LB, 1EX	D
5	E	Excavation stage 2	3	2LB, 1EX	-
6	F	Formwork stage 2	3	4LB, 1FM, 1MC	E
7	G	Concrete stage 2	6	4LB	G
8	H	Backfill stage 2	2	2LB, 1EX	H
9	I	Erect steel work	3	3LB, 2MC, 1ST	D, H

Similar to case study 1, the four hyperparameters, namely the number of episodes, memory capacity, number of steps to update GNN, and batch size, were set to 4000, 10,000, 2, and 16, respectively.

The result of the AON network using the proposed RL-ABM method for case study 2 is shown in Fig. 10, and the corresponding Gantt chart is shown in Fig. 11. Given the resources assigned, the footbridge construction took 24 days to complete and the prefabricated superstructure was ready for erection by day 21. Compared with previous research referenced for this case study [77], RL-ABM reduces total project finish time by 3 days.

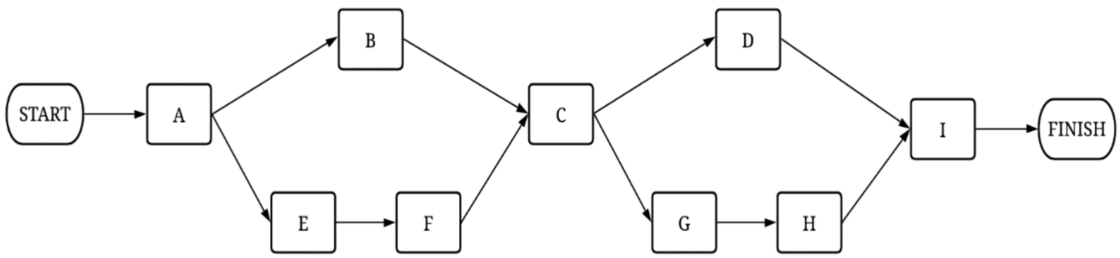

Fig. 10. Resulting AON network for case study 2 based on the proposed RL-ABM method.

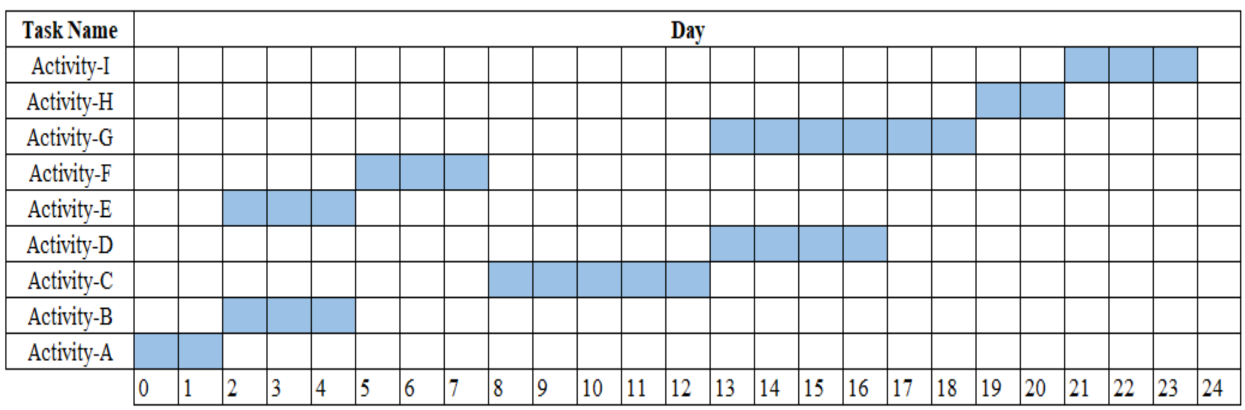

Fig. 11. Corresponding Gantt chart for the RL-ABM solution for case study 2.

4.3. Case study 3

The third case study is a scheduling problem adapted from Zhang et al. [78]. The case study was specifically selected in order to compare RL-ABM output with the results presented in the aforementioned research [78]. In effect, results of RL-ABM output are compared with from other heuristic methods, namely: minimum total float (MITF), shortest activity duration (SAD), minimum late finish time (MILFT), genetic algorithm (GA), and particle swarm optimization (PSO). In this case study [78], the scheduling problem consists of 25 activities and 2 dummy activities, with three different types of resources (R1, R2, and R3). The AON network is illustrated in Fig. 12., and the structure of the activity table is shown in Table 4.

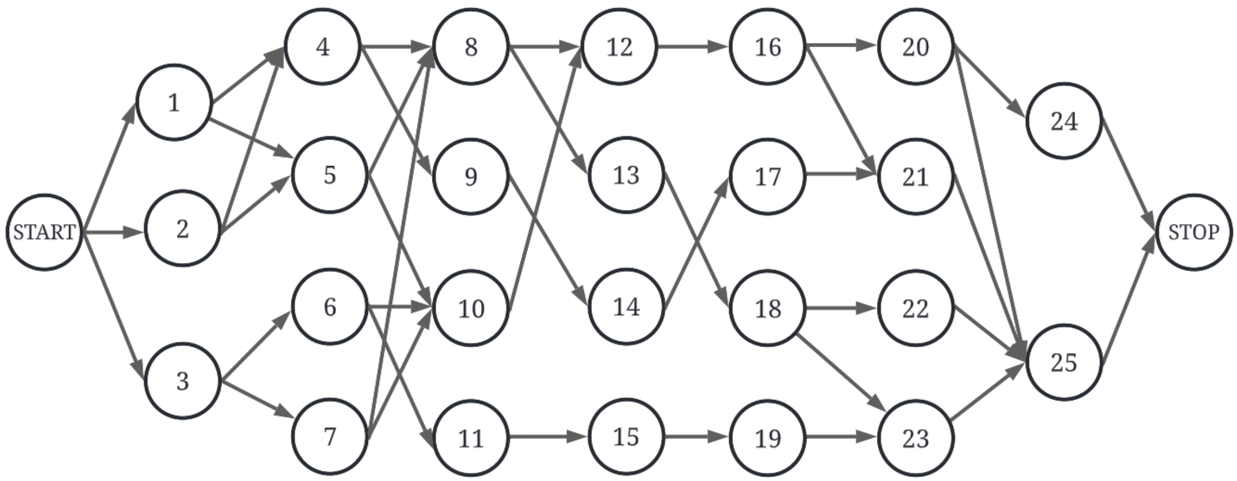

Fig. 12. AON network of case-study 3, a more complicated scheduling problem (adapted from[78]).

Table 4. Structure of activity table for case study 3, a more complicated scheduling problem (adapted from[78]).

Node Number	Activity	Duration	Resources	Predecessor(s)
1	1	5	{'r1': '5', 'r2': '3', 'r3': '2'}	-
2	2	5	{'r1': '4', 'r2': '5', 'r3': '3'}	-
3	3	3	{'r1': '2', 'r2': '5', 'r3': '2'}	-
4	4	4	{'r1': '1', 'r2': '4', 'r3': '4'}	1, 2
5	5	2	{'r1': '4', 'r2': '2', 'r3': '4'}	1, 2
6	6	1	{'r1': '5', 'r2': '5', 'r3': '4'}	3
7	7	6	{'r1': '5', 'r2': '3', 'r3': '2'}	3
8	8	6	{'r1': '2', 'r2': '3', 'r3': '2'}	4, 5, 7
9	9	1	{'r1': '1', 'r2': '4', 'r3': '4'}	4
10	10	3	{'r1': '2', 'r2': '3', 'r3': '4'}	5, 6, 7
11	11	3	{'r1': '3', 'r2': '3', 'r3': '2'}	6
12	12	3	{'r1': '4', 'r2': '1', 'r3': '4'}	8, 10
13	13	3	{'r1': '5', 'r2': '5', 'r3': '4'}	8
14	14	6	{'r1': '2', 'r2': '2', 'r3': '2'}	9
15	15	4	{'r1': '5', 'r2': '1', 'r3': '4'}	11
16	16	3	{'r1': '3', 'r2': '5', 'r3': '3'}	12
17	17	3	{'r1': '2', 'r2': '3', 'r3': '3'}	14
18	18	4	{'r1': '5', 'r2': '4', 'r3': '4'}	13, 15
19	19	1	{'r1': '4', 'r2': '2', 'r3': '6'}	15
20	20	4	{'r1': '0', 'r3': '4', 'r2': '1'}	16

21	21	4	{'r1': '6', 'r2': '1', 'r3': '2'}	16, 17
22	22	1	{'r1': '2', 'r2': '2', 'r3': '1'}	18
23	23	6	{'r1': '2', 'r2': '3', 'r3': '1'}	18, 19
24	24	3	{'r1': '2', 'r2': '2', 'r3': '2'}	20
25	25	3	{'r1': '1', 'r2': '0', 'r3': '3'}	20, 21, 22, 23

The four hyperparameters utilized for modeling this problem, namely, the number of episodes, memory capacity, number of steps to update GNN, and batch size, were set to 4000, 10,000, 2, and 16, respectively. These model parameters are similar to those for case studies 1 and 2.

The result of the RL-ABM output is shown in the Gantt chart in Fig. 13. The total duration for the project is computed to be 64 days, where different allocation is assigned for the three resources shown in Figs. 14–16.

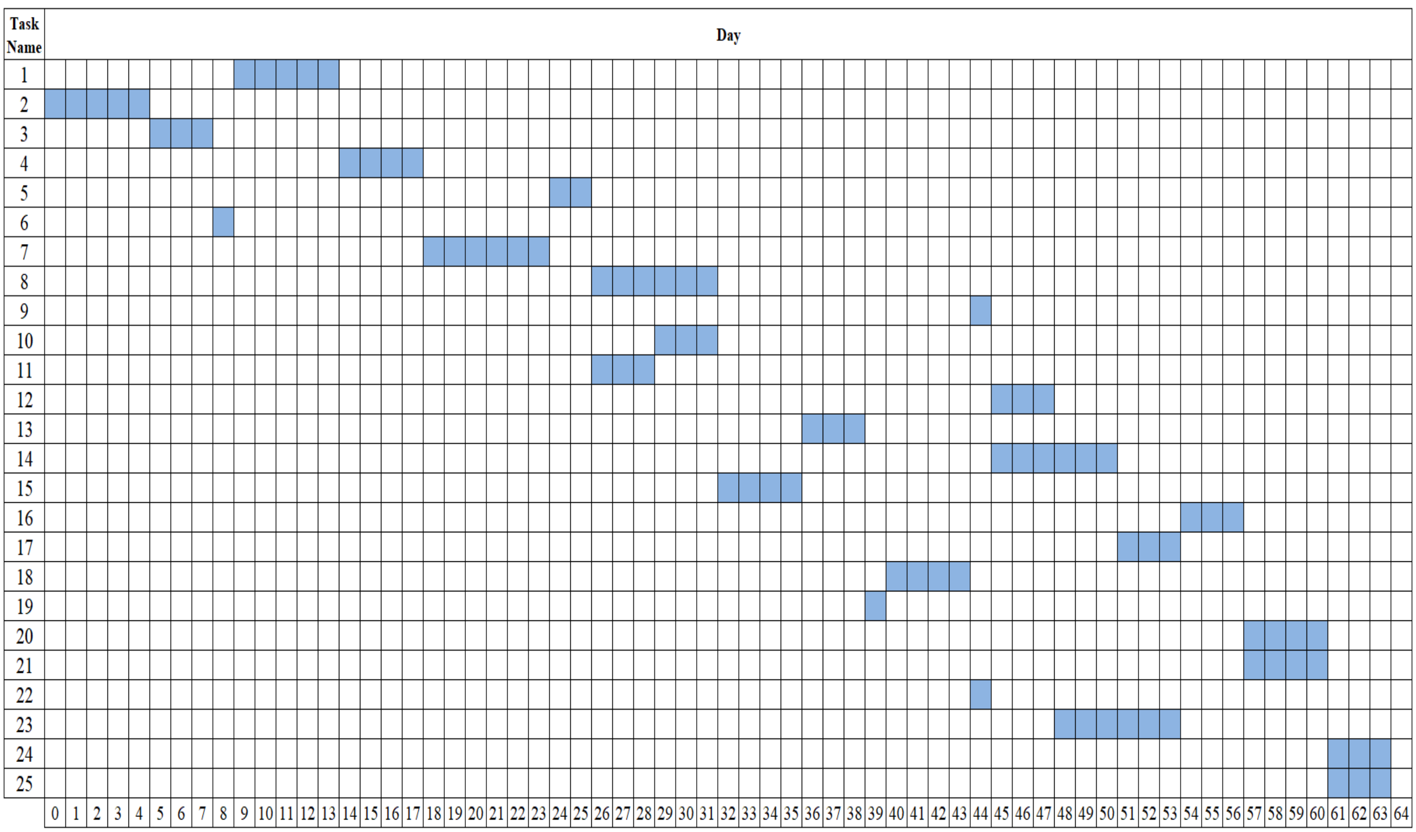

Fig. 13. Corresponding Gantt chart for the RL-ABM solution for case study 3.

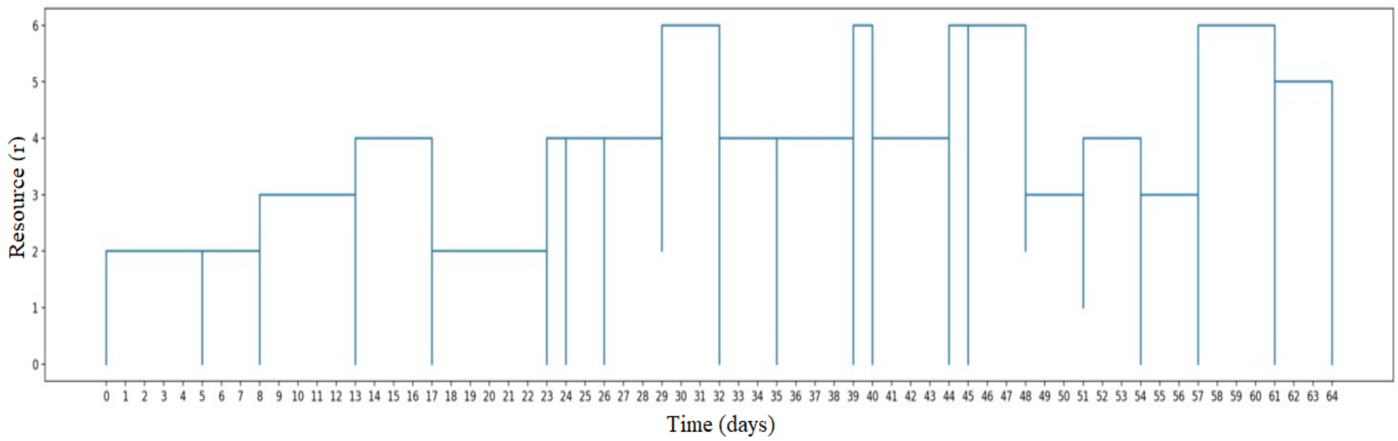

Fig. 14. Model output: resource profile for R1.

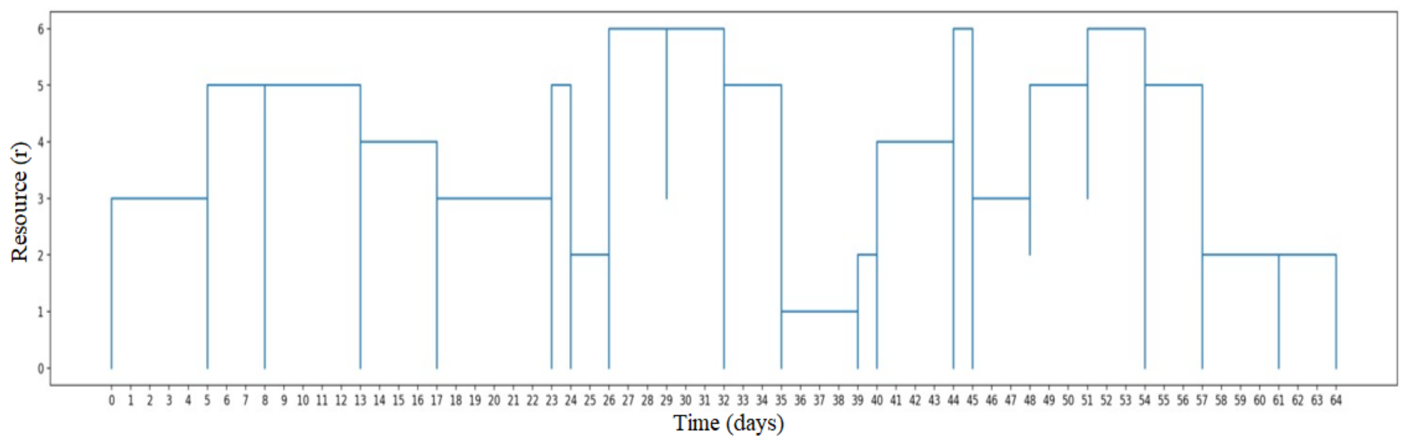

Fig. 15. Model output: resource profile for R2.

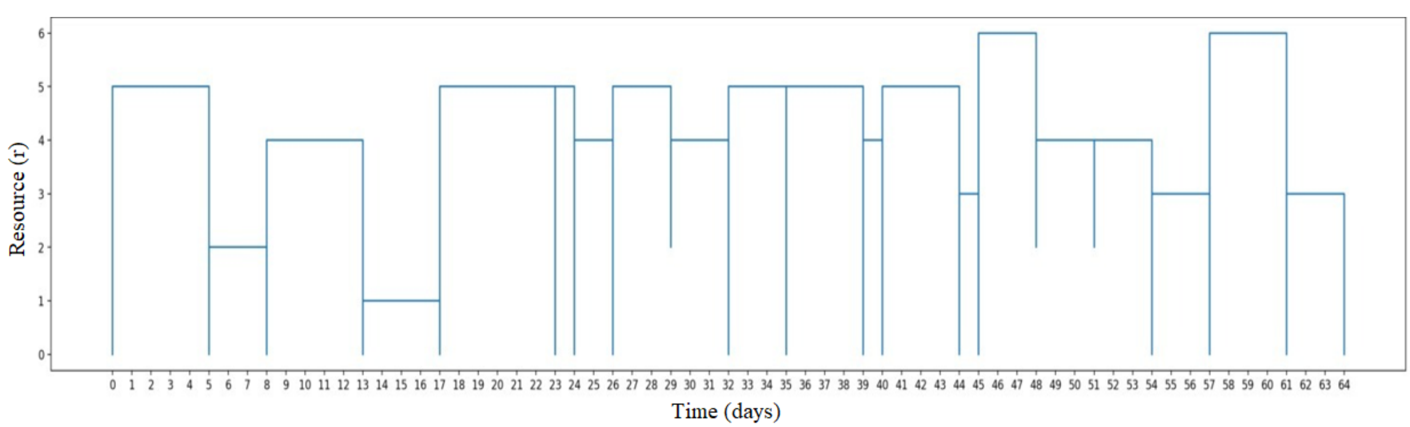

Fig. 16. Model output: resource profile for R3.

4.4. Discussion

This section discusses the validation of the proposed RL-ABM method in each case study and insights into how construction practitioners can utilize results of the proposed model to optimize their scheduling problems.

In case study 1, the ABM process included 4000 iterations of forward and backward passes from “NotReady to “Complete,” as shown in Fig. 4, to obtain the RL agent parameters. Accordingly, the RL agent checked whether enough labor resource was available for an activity in the "Ready" state. After the RL agent identified the sufficient labor resource required for the activity, the ABM simulation environment started the “InProgress” and then “Completed” states. As a result, the model shows a minor training loss with low computational average time. Specifically, Fig. 17 shows the training loss for 4000 iterations. In effect, after reducing ε to less than 0.5, the policy converges to optimum value. The processing efficiency is described in terms of the time taken to execute the model. The model was run using a desktop computer, Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, and took less than 2 minutes to complete. Fig. 18 shows the learning loss and average time for each episode.

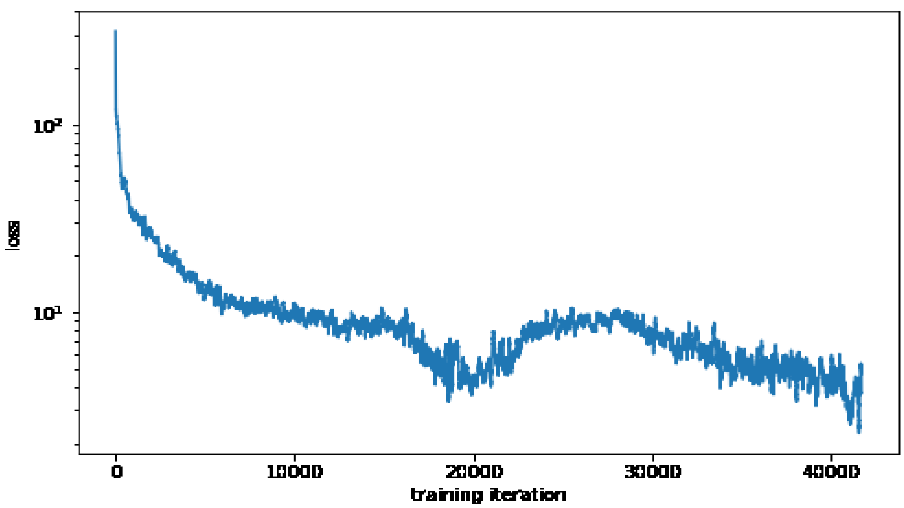

Fig. 17. Training loss for 4000 iterations in case study 1.

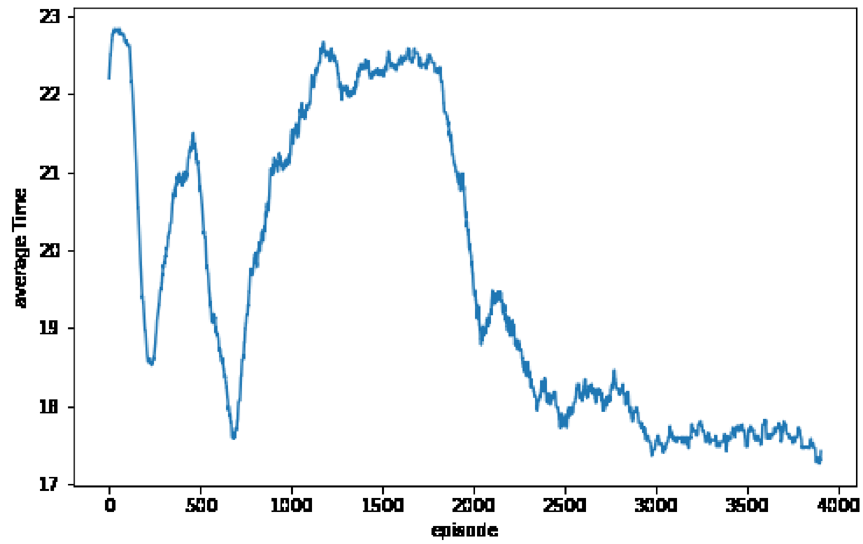

Fig. 18. Average time for 4000 episodes in case study 1.

In case study 2, similarly using 4000 iterations in the ABM simulation process with a more complex set of resources, the RL-ABM method also shows an efficient optimization outcome. Accordingly, there is low training loss in the ABM simulation after the RL agent identifies the resources to change from “NotReady to “Complete.” Fig. 19 shows the learning loss for each episode. Convergence of policy to optimum value is obtained after reducing ε to less than 0.5, as Fig. 17, Fig. 19, and Fig. 21. The processing efficiency is described in terms of time taken to execute the model. Fig. 20 gives the average time per episode, which shows a decreasing trend at the end of episode 4000.

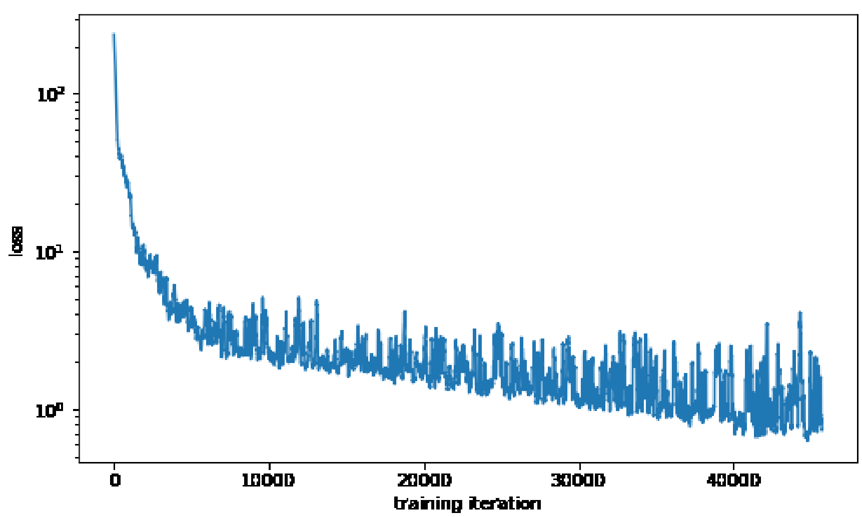

Fig. 19. Training loss for 4000 iterations in case study 2.

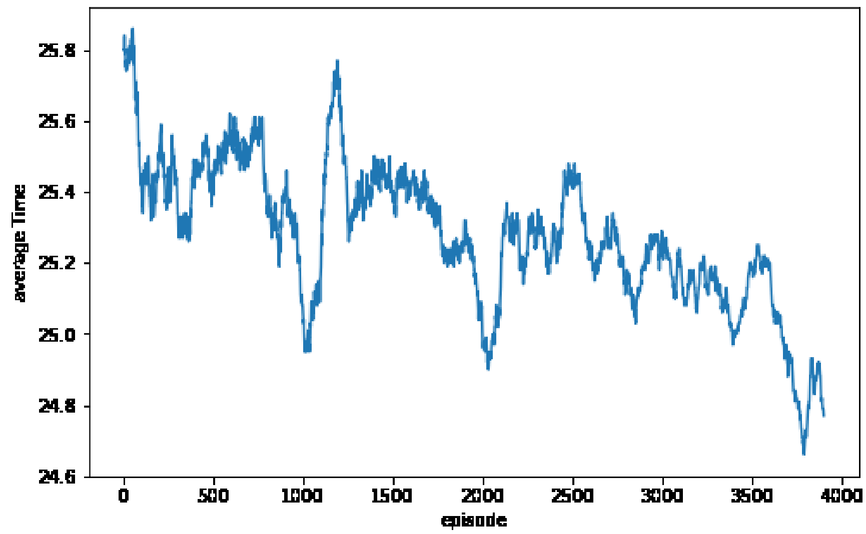

Fig. 20. Average time for 4000 episodes in case study 2.

In case study 3, the model shows a similar minor training loss with low computational average time as the previous two case studies do under the same 4000 iterations in the ABM simulation environment. Fig. 21 shows the training loss for 4000 iterations. The processing efficiency is described in terms of the time taken to execute the model. Fig. 22 shows the learning loss and average time for each episode.

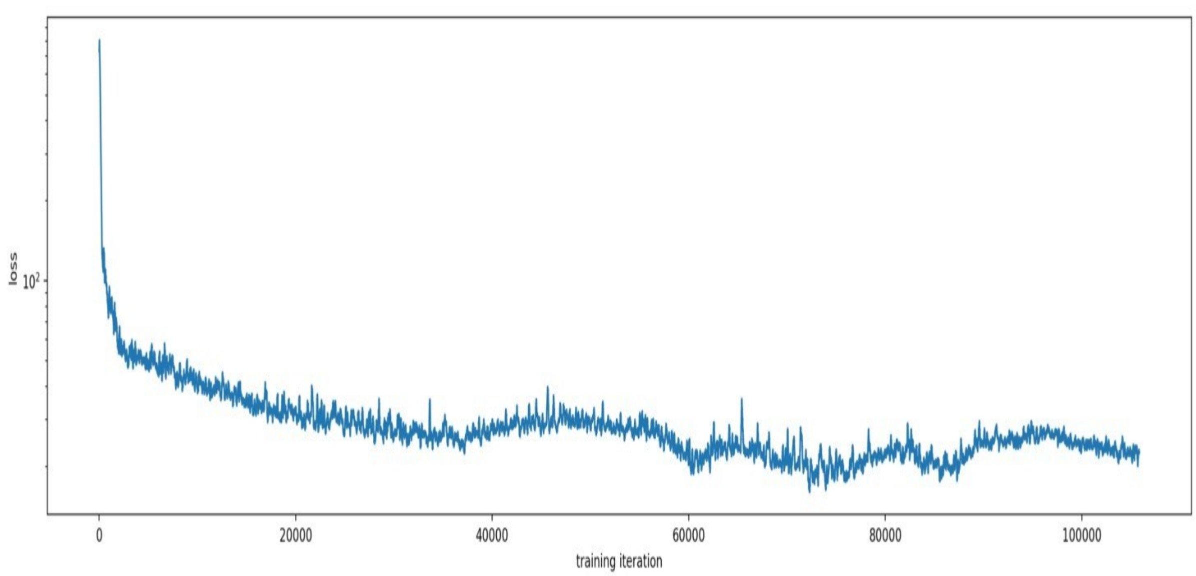

Fig. 21. Training loss for 4000 iterations in case study 3.

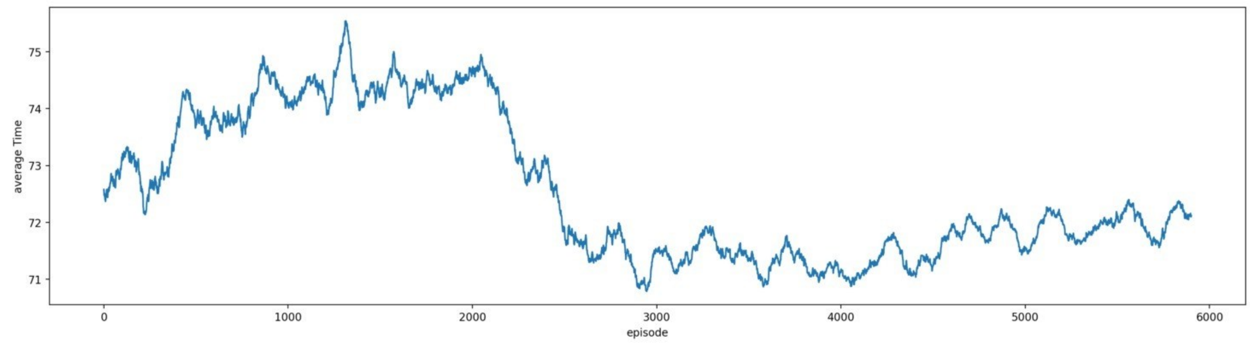

Fig. 22. Average time for 4000 episodes in case study 3.

The hybridization of RL-ABM and graph embedding methods proposed in this study elucidates advanced machine learning techniques that can be used in construction scheduling optimization. The case study results indicate that compared to the solution proposed using the resource-constrained CPM scheduling, outcomes from the proposed RL-ABM method provide greater improvements in optimizing project durations. In case study 1, RL-ABM improved the total project duration by 15 percent. Similarly, results from case study 2 show an improvement of 15 percent in project duration. Moreover, the results from case study 3, which show a more complicated set of activities with a greater resource profile, demonstrate the capability of RL-ABM to address more complicated problems and produce comparable results with better efficiency and that RL-ABM performed better compared with the results from other heuristic approaches in Zhang et al. [78]. The results from MITF, SAD, and MILFT produced 74 days, 71 days, and 67 days, respectively. Compared with the results from GA and PSO, RL-ABM had a similar result of 64 days. However, the proposed RL-ABM method offers a significantly greater advantage, not only because of its computational efficiency, but also because it is able to provide several scheduling scenarios where the minimum possible duration can be reached. In this regard, RL-ABM offers multiple scenarios of scheduling to achieve the minimum duration, where planners can make activity sequencing decisions based on other additional criteria, such as resource leveling.

The case study results also indicate that the proposed RL-ABM method provides a more comprehensive approach to planning, because it provides a dynamic solution to the optimization problem by effectively changing the AON network even as project situations change on the construction site. This feature makes the model capable of proposing flexible planning solutions in changing construction environments, such as adapting initial WBS and AON when project conditions change. This paper also extends the application of RL-ABM for proposing construction planning solutions by incorporating the graph-embedding method to enable handling of more activities and activity network relationships for use in the RL optimization platform.

Furthermore, the RL-ABM method proposed in this study has an accompanying user-friendly application that allows practitioners to utilize this model using easy-to-understand features embedded in a graphical user interface (GUI). The different sections of the application are shown in Fig. 23, Fig. 24, Fig. 25, and Fig. 26, which detail the simple steps a user has to perform to utilize RL-ABM.

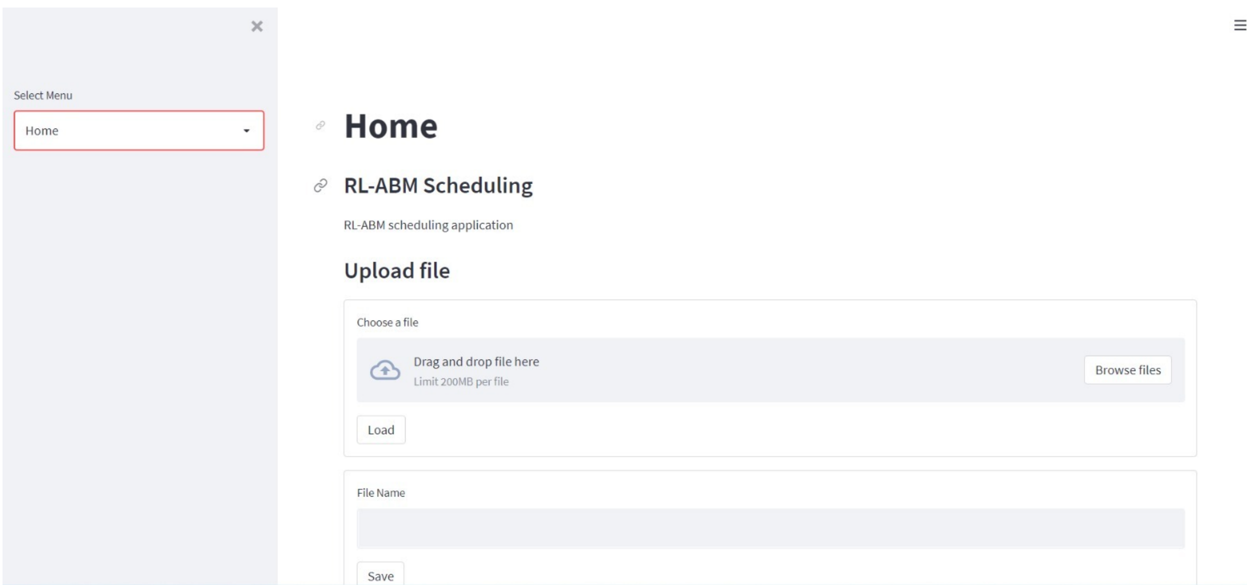

Fig. 23. Prompt to enter scheduling data (precedence relationships) from a file.

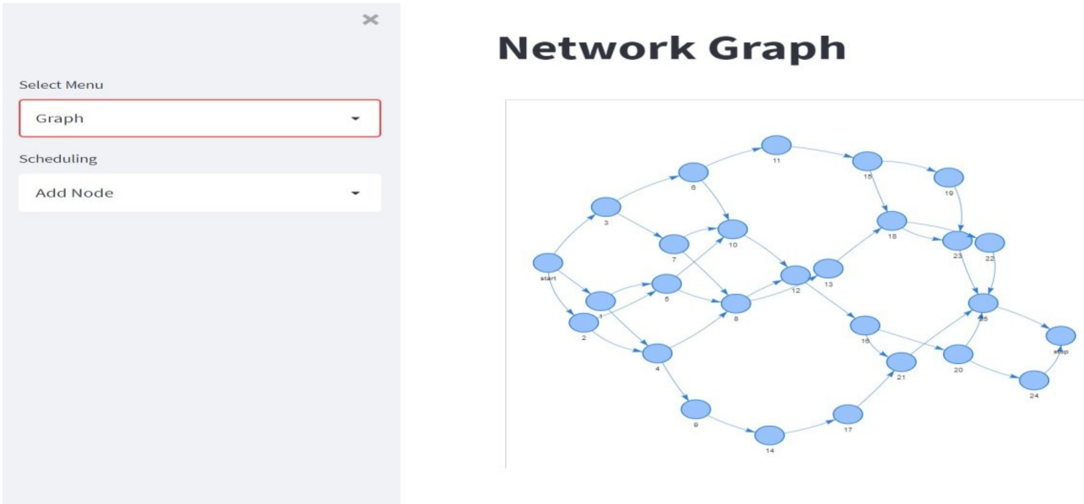

Fig. 24. Prompt to add node for new activities.

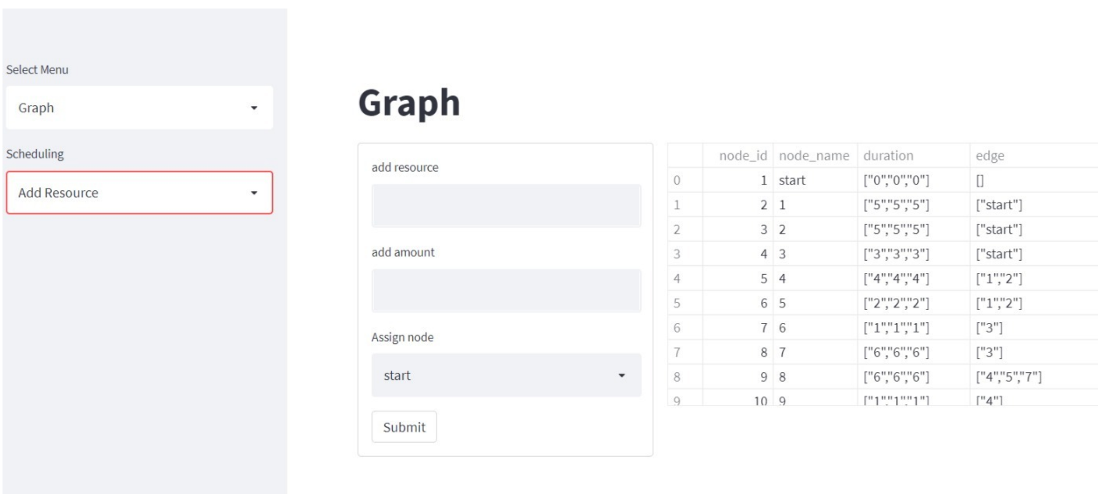

Fig. 25. Prompt to log resource requirements.

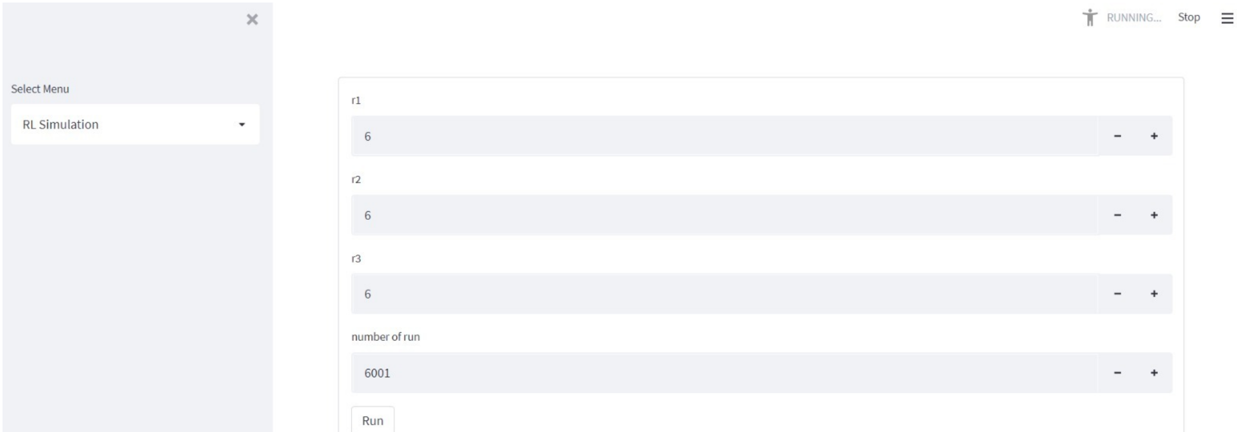

Fig. 26. Interface to execute the RL-ABM method and view iterations.

As Fig. 23 illustrates, the user needs to first input the precedence relationships for each activity, which includes the duration of each activity in relation to the associated technological constraints. This is performed by reading a dataset created as an Excel or .csv file format. The user is also able to add nodes using the GUI feature, as shown in Fig. 24. Next, the user is prompted to input resource requirements corresponding to each activity and assign nodes, as shown in Fig. 25. Finally, as shown in Fig. 26, the 'Run' button enables the user to execute the model and get the results based on predefined RL-ABM parameters.

5. Conclusions and Future Work

In construction planning, the optimal solution for sequencing activities is often selected from a set of finite solutions. However, the optimization problem is everchanging, because the environment, which includes the number of activities, type, and number of allocated resources, changes during execution of the project. Agents in RL algorithms learn better solutions even as the environment changes. A review of the literature emphasizes the need for an effective decision-making tool that can be easily used by stakeholders in accordance with their preferences for improving project performance with respect to constraints such as time, cost, and quality.

This study developed a hybrid RL-ABM method to support decision-making in construction planning that includes three major steps: converting a construction schedule to a graph network, performing ABM, and implementing RL to perform schedule optimization. The proposed model was demonstrated using three case studies in construction scheduling problems obtained from the work of Lu and Li [77] and Zhang et al. [78]. As a result, using ABM was shown to better enable representation of the construction environment through the use of state charts. This is because complex relationships, which are the function of an activity's parameters, including an activity's lifespan from “Started” to “Finished,” as well as agent interactions, including activities competition to obtain resources could be effectively captured with the principle of ABM.

This study has some limitations. First, the underlying uncertainties related to activity duration were not demonstrated in this manuscript despite the proposed model having such features, as more focus was given to presenting how the model works compared with other previous similar studies. The study is also limited to addressing single-objective optimization (minimize project duration) subject to single or multiple constraints. Optimization of multiple objectives using multiple RL-agents was not performed in this study. In future work, the proposed RL-ABM method will be extended to represent a more comprehensive project by incorporating varying distributions of project duration, multiple sub-contractors, and varying descriptions of resources including equipment specification, labor profile, and experience during the simulation process. Moreover, the proposed model will also be extended to perform multi-objective optimizations with more constraints, such as time, cost, and quality, by incorporating multiple RL agents. Using multi-agent reinforcement learning approach, conflicting objectives such as increasing direct cost versus minimizing duration will be addressed by using mixed cooperative–competitive RL-agents in the RL platform. The RL-ABM GUI will also be improved to include an interface that enables changing RL-ABM hyperparameters, which can allow the user to become more seamlessly involved in the training process.

Data Availability Statement

All data, models, and code generated or used during the study appear in the published article.

Acknowledgement

This research is funded by the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair in Strategic Construction Modeling and Delivery (NSERC IRCPJ 428226–15), which is held by Dr. Aminah Robinson Fayek.

References

[1] A. Laufer, D. Cohenca, Factors affecting construction-planning outcomes, Journal of Construction Engineering and Management 116 (1) (1990) pp. 135–156. https://doi.org/10.1061/(asce)0733-9364(1990)116:1(135).

[2] PMI, A guide to the project management body of knowledge (PMBOK® Guide)-Fifth edition, Project Management Journal 44 (3) (2013). https://doi.org/10.1002/pmj.21345.

[3] H. Muñoz-Avila, K. Gupta, D.W. Aha, D. Nau, Knowledge-based project planning, in: R. Dieng-Kuntz, N. Matta (Eds.), Knowledge Management and Organizational Memories, Springer, Boston, MA, USA, 2002: pp. 125–134. https://doi.org/10.1007/978-1-4615-0947-9_11.

[4] F. Amer, H.Y. Koh, M. Golparvar-Fard, Automated methods and systems for construction planning and scheduling: Critical review of three decades of research, Journal of Construction Engineering and Management 147 (7) (2021) 03121002. https://doi.org/10.1061/(asce)co.1943-7862.0002093.

[5] E. Ratajczak-Ropel, Experimental evaluation of agent-based approaches to solving multi-mode resource-constrained project scheduling problem, Cybernetics and Systems 49 (5–6) (2018) pp. 296–316. https://doi.org/10.1080/01969722.2017.1418269.

[6] R.K. Soman, M. Molina-Solana, Automating look-ahead schedule generation for construction using linked-data based constraint checking and reinforcement learning, Automation in Construction 134 (2022) 104069. https://doi.org/10.1016/j.autcon.2021.104069.

[7] M. Raoufi, A.R. Fayek, Fuzzy agent-based modeling of construction crew motivation and performance, Journal of Computing in Civil Engineering 32 (5) (2018). https://doi.org/10.1061/(asce)cp.1943-5487.0000777.

[8] M.A. Abdelmegid, V.A. González, M. Poshdar, M. O'Sullivan, C.G. Walker, F. Ying, Barriers to adopting simulation modelling in construction industry, Automation in Construction 111 (2020) 103046. https://doi.org/10.1016/j.autcon.2019.103046.

[9] W.K.V. Chan, Y.J. Son, C.M. Macal, Agent-based simulation tutorial - Simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation, in: Proceedings - Winter Simulation Conference, IEEE, 2010: pp. 135–150. https://doi.org/10.1109/WSC.2010.5679168.

[10] M. Watkins, A. Mukherjee, N. Onder, K. Mattila, Using agent-based modeling to study construction labor productivity as an emergent property of individual and crew interactions, Journal of Construction Engineering and Management 135 (7) (2009). https://doi.org/10.1061/(asce)co.1943-7862.0000022.

[11] N.S. Kedir, M. Raoufi, A.R. Fayek, Fuzzy agent-based multicriteria decision-making model for analyzing construction crew performance, Journal of Management in Engineering 36 (5) (2020) 04020053. https://doi.org/10.1061/(asce)me.1943-5479.0000815.

[12] M.N. Bakht, T.E. El-Diraby, Synthesis of decision-making research in construction, Journal of Construction Engineering and Management 141 (9) (2015) 04015027. https://doi.org/10.1061/(asce)co.1943-7862.0000984.

[13] D. Jato-Espino, E. Castillo-Lopez, J. Rodriguez-Hernandez, J.C. Canteras-Jordana, A review of application of multi-criteria decision making methods in construction, Automation in Construction 45 (2014) pp. 151–162. https://doi.org/10.1016/j.autcon.2014.05.013.

[14] J. Zhou, P.E.D. Love, X. Wang, K.L. Teo, Z. Irani, A review of methods and algorithms for optimizing construction scheduling, Journal of the Operational Research Society 64 (8) (2013) pp. 1091–1105. https://doi.org/10.1057/jors.2012.174.

[15] M. Alemi-Ardakani, A.S. Milani, S. Yannacopoulos, G. Shokouhi, On the effect of subjective, objective and combinative weighting in multiple criteria decision making: A case study on impact optimization of composites, Expert Systems with Applications 46 (2016) pp. 426–438. https://doi.org/10.1016/j.eswa.2015.11.003.

[16] T. Hegazy, Construction progress control, in: Computer-Based Construction Project Management, Prentice Hall PTR, Hoboken, New Jersey/Ohio, 2002: pp. 289–316. ISBN: 978-0130888594.

[17] M.-F.F. Siu, M. Lu, S. AbouRizk, Resource supply-demand matching scheduling approach for construction workface planning, Journal of Construction Engineering and Management 142 (1) (2016). https://doi.org/10.1061/(asce)co.1943-7862.0001027.

[18] A. Sonmez, E. Kocyigit, E. Kugu, Optimal path planning for UAVs using Genetic Algorithm, in: 2015 International Conference on Unmanned Aircraft Systems (ICUAS 2015), 2015, pp. 50–55. https://doi.org/10.1109/ICUAS.2015.7152274.

[19] M. Yahya, M.P. Saka, Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights, Automation in Construction 38 (2014) pp. 14–29. https://doi.org/10.1016/j.autcon.2013.11.001.

[20] Y. Liu, C. Chu, K. Wang, A new heuristic algorithm for the operating room scheduling problem, Computers and Industrial Engineering 61 (3) (2011) pp. 865–871. https://doi.org/10.1016/j.cie.2011.05.020.

[21] P.H. Chen, S.M. Shahandashti, Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints, Automation in Construction 18 (4) (2009) pp. 434–443. https://doi.org/10.1016/j.autcon.2008.10.007.

[22] T. Hegazy, M. Kassab, Resource optimization using combined simulation and genetic algorithms, Journal of Construction Engineering and Management 129 (6) (2003) pp. 698–705. https://doi.org/10.1061/(asce)0733-9364(2003)129:6(698).

[23] A.G. Correia, M. Parente, P. Cortez, Earthwork optimization system for sustainable highway construction, in: A.O. Sfriso, D. Manzanal, R.J. Rocca (Eds.), Advances in Soil Mechanics and Geotechnical Engineering Series, Volume 5: Geotechnical Synergy, I0S Books, Buenos Aires, 2015: pp. 121–138. https://doi.org/10.3233/978-1-61499-599-9-121.

[24] S.A.H. Golpayegani, F. Parvaresh, The logical precedence network planning of projects, considering the finish-to-start (FS) relations, using neural networks, International Journal of Advanced Manufacturing Technology 55 (9) (2011) pp. 1123–1133. https://doi.org/10.1007/s00170-010-3125-1.

[25] E. Mikulakova, M. König, E. Tauscher, K. Beucke, Knowledge-based schedule generation and evaluation, in: Advanced Engineering Informatics, 24 (4) (2010): pp. 389–403. https://doi.org/10.1016/j.aei.2010.06.010.

[26] A. Slowik, H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Computing and Applications. 32(16) (2020) pp. 12363–12379. https://doi.org/10.1007/s00521-020-04832-8.

[27] M.H. Nili, H. Taghaddos, B. Zahraie, Integrating discrete event simulation and genetic algorithm optimization for bridge maintenance planning, Automation in Construction 122 (2021) 103513. https://doi.org/10.1016/j.autcon.2020.103513.

[28] K. Feng, S. Chen, W. Lu, Machine learning based construction simulation and optimization, in: Proceedings - Winter Simulation Conference, IEEE, 2018: pp. 2025–2036. https://doi.org/10.1109/WSC.2018.8632290.

[29] S. RazaviAlavi, S. AbouRizk, Site layout and construction plan optimization using an integrated genetic algorithm simulation framework, Journal of Computing in Civil Engineering 31 (4) (2017). https://doi.org/10.1061/(asce)cp.1943-5487.0000653.

[30] P.M. Teicholz, A simulation approach to the selection of construction equipment, Technical Report No. 26, 1963. https://www.worldcat.org/title/simulation-approach-to-the-selection-of-construction-equipment/oclc/020027256 (accessed May 2, 2022).

[31] G. Gordon, A general purpose systems simulation program, in: Proceedings of the Eastern Joint Computer Conference: Computers - Key to Total Systems Control (AFIPS), 1961: pp. 87–104. https://doi.org/10.1145/1460764.1460768.

[32] M. Lu, Simplified discrete-event simulation approach for construction simulation, Journal of Construction Engineering and Management 129 (5) (2003) pp. 537–546. https://doi.org/10.1061/(asce)0733-9364(2003)129:5(537).

[33] A. Alvanchi, R. Azimi, S. Lee, S.M. AbouRizk, P. Zubick, Off-site construction planning using discrete event simulation, Journal of Architectural Engineering 18 (2) (2012) pp. 114–122. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000055.

[34] H. Liu, M.S. Altaf, M. Lu, Automated production planning in panelized construction enabled by integrating discrete-event simulation and BIM, in: The Canadian Society for Civil Engineering 5th International Construction Specialty Conference, 2015: pp. 1–10. https://www.researchgate.net/profile/Hexu-Liu/publication/279200553_Automated_production_planning_in_panelized_construction_enabled_by_integrating_discrete-event_simulation_and_BIM/links/558f779808ae1e1f9bacec5b/Automated-production-planning-in-panelized-construction-enabled-by-integrating-discrete-event-simulation-and-BIM.pdf?origin=publication_detail (accessed May 3, 2022).

[35] H. Zhang, Discrete-event simulation for estimating emissions from construction processes, Journal of Management in Engineering 31 (2) (2015). https://doi.org/10.1061/(asce)me.1943-5479.0000236.

[36] C. Ahn, W. Pan, S.H. Lee, F. Peña-Mora, Enhanced estimation of air emissions from construction operations based on discrete-event simulation, in: W. Tizani (Ed.), Proceedings of the 2010 International Conference on Computing in Civil and Building Engineering (ICCBE) 2010 and 17th International Workshop on Intelligent Computing in Engineering, University of Nottingham, UK, 2010. http://www.engineering.nottingham.ac.uk/icccbe/proceedings/pdf/pf119.pdf.

[37] H.D. Khanh, S.Y. Kim, Exploring productivity of concrete truck for multistory building projects using discrete event simulation, KSCE Journal of Civil Engineering. 24 (2020) pp. 3531–3545. https://doi.org/10.1007/s12205-020-1389-z.

[38] K.P. Kisi, N. Mani, E.M. Rojas, E.T. Foster, Optimal productivity in labor-intensive construction operations: Pilot study, Journal of Construction Engineering and Management 143 (2) (2017) 04016107. https://doi.org/10.1061/(asce)co.1943-7862.0001257.

[39] M. Arashpour, M. Arashpour, Analysis of workflow variability and its impacts on productivity and performance in construction of multistory buildings, Journal of Management in Engineering 31 (6) (2015) 04015006. https://doi.org/10.1061/(asce)me.1943-5479.0000363.

[40] K.M. Shawki, K. Kilani, M.A. Gomaa, Analysis of earth-moving systems using discrete-event simulation, Alexandria Engineering Journal 54 (3) (2015) pp. 533–540. https://doi.org/10.1016/j.aej.2015.03.034.

[41] J.C. Martinez, Methodology for conducting discrete-event simulation studies in construction engineering and management, Journal of Construction Engineering and Management 136 (1) (2010) pp. 3–16. https://doi.org/10.1061/(asce)co.1943-7862.0000087.

[42] E. Zankoul, H. Khoury, R. Awwad, Evaluation of agent-based and discrete-event simulation for modeling construction earthmoving operations, in: Proceedings of the 32nd International Symposium on Automation and Robotics (ISARC) in Construction and Mining: Connected to the Future, 2015: pp. 1–8. https://doi.org/10.22260/isarc2015/0014.

[43] C.M. Macal, M.J. North, Agent-based modeling and simulation: Desktop ABMS, in: Proceedings - Winter Simulation Conference, IEEE, 2007: pp. 95–106. https://doi.org/10.1109/WSC.2007.4419592.

[44] M.P. Wellman, Putting the agent in agent-based modeling, Autonomous Agents and Multi-Agent Systems 30 (6) (2016) pp. 1175–1189. https://doi.org/10.1007/s10458-016-9336-6.

[45] J.C.P. Cheng, Y. Tan, Y. Song, Z. Mei, V.J.L. Gan, X. Wang, Developing an evacuation evaluation model for offshore oil and gas platforms using BIM and agent-based model, Automation in Construction 89 (2018) pp. 214–224. https://doi.org/10.1016/j.autcon.2018.02.011.

[46] A. Jabri, T. Zayed, Agent-based modeling and simulation of earthmoving operations, Automation in Construction 81 (2017) pp. 210–223. https://doi.org/10.1016/j.autcon.2017.06.017.

[47] Y. Cao, T. Wang, X. Song, An energy-aware, agent-based maintenance-scheduling framework to improve occupant satisfaction, Automation in Construction 60 (2015) pp. 49–57. https://doi.org/10.1016/j.autcon.2015.09.002.

[48] M.S. Eid, I.H. El-adaway, Decision-making framework for holistic sustainable disaster recovery: Agent-based approach for decreasing vulnerabilities of the associated communities, Journal of Infrastructure Systems 24 (3) (2018) 04018009. https://doi.org/10.1061/(asce)is.1943-555x.0000427.

[49] F. Nasirzadeh, M. Khanzadi, M. Mir, A hybrid simulation framework for modelling construction projects using agent-based modelling and system dynamics: an application to model construction workers' safety behavior, International Journal of Construction Management 18 (2) (2018) pp. 132–143. https://doi.org/10.1080/15623599.2017.1285485.

[50] D.D. Wu, X. Kefan, L. Hua, Z. Shi, D.L. Olson, Modeling technological innovation risks of an entrepreneurial team using system dynamics: An agent-based perspective, Technological Forecasting and Social Change 77 (6) (2010) pp. 857–869. https://doi.org/10.1016/j.techfore.2010.01.015.

[51] Z. Zhang, J. Yang, H. Zha, Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization, in: B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (Eds.), Proceedings of the 19th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2020: pp. 2083-2085. https://www.ifaamas.org/Proceedings/aamas2020/pdfs/p2083.pdf (accessed May 3, 2022).

[52] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, Second edition, Cambridge University Press, 2018. ISBN: 978-0262039246.

[53] V. Shitole, J. Louis, P. Tadepalli, Optimizing earth moving operations via reinforcement learning, in: Proceedings - Winter Simulation Conference, 2019: pp. 2954–2965. https://doi.org/10.1109/WSC40007.2019.9004935.

[54] F. Bertoni, M. Giuliani, A. Castelletti, Integrated design of dam size and operations via reinforcement learning, Journal of Water Resources Planning and Management 146 (4) (2020) 04020010. https://doi.org/10.1061/(asce)wr.1943-5452.0001182.

[55] B. Bhattacharya, A.H. Lobbrecht, D.P. Solomatine, Neural networks and reinforcement learning in control of water systems, Journal of Water Resources Planning and Management 129 (6) (2003) pp. 458–465. https://doi.org/10.1061/(asce)0733-9496(2003)129:6(458).

[56] W. Genders, S. Razavi, Asynchronous n-step Q-learning adaptive traffic signal control, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 23 (4) (2019) pp. 319–331. https://doi.org/10.1080/15472450.2018.1491003.

[57] B. Yin, M. Menendez, A reinforcement learning method for traffic signal control at an isolated intersection with pedestrian flows, in: Proceedings of the 19th COTA International Conference of Transportation Professionals (CICTP), Nanjing, China, 2019. https://doi.org/10.1061/9780784482292.270.

[58] J.C. Medina, R.F. Benekohal, Reinforcement learning agents for traffic signal control in oversaturated networks, in: Proceedings of the First Transportation and Development Institute (TDI), 2011: pp. 132–141. https://doi.org/10.1061/41167(398)14.

[59] P.L. Durango, Reinforcement learning models for transportation infrastructure management, in: K.C.P. Wang, S. Madanat, S. Nambisan, G. Spring (Eds.), Proceedings of the Seventh International Conference on Applications of Advanced Technologies in Transportation (AATT), 2002: pp. 568–575. https://doi.org/10.1061/40632(245)72.

[60] G.H. Erharter, T.F. Hansen, Z. Liu, T. Marcher, Reinforcement learning based process optimization and strategy development in conventional tunneling, Automation in Construction 127 (2021) 103701. https://doi.org/10.1016/j.autcon.2021.103701.

[61] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Applied Energy 269 (2020) 115036. https://doi.org/10.1016/j.apenergy.2020.115036.

[62] Z. Zhang, A. Chong, Y. Pan, C. Zhang, K.P. Lam, Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning, Energy and Buildings 199 (2019) pp. 472–490. https://doi.org/10.1016/j.enbuild.2019.07.029.

[63] S. Zhou, Z. Hu, W. Gu, M. Jiang, X.-P. Zhang, Artificial intelligence based smart energy community management: A reinforcement learning approach, CSEE Journal of Power and Energy Systems 5 (1) (2019) pp. 1–10. https://doi.org/10.17775/CSEEJPES.2018.00840.

[64] F. Ruelens, B.J. Claessens, S. Vandael, B. de Schutter, R. Babuska, R. Belmans, Residential demand response of thermostatically controlled loads using batch reinforcement learning, IEEE Transactions on Smart Grid 8 (5) (2017) pp. 2149–2159. https://doi.org/10.1109/TSG.2016.2517211.

[65] H. Berlink, N. Kagan, A.H. Reali Costa, Intelligent decision-making for smart home energy management, Journal of Intelligent and Robotic Systems: Theory and Applications 80 (1) (2015) pp. 331–354. https://doi.org/10.1007/s10846-014-0169-8.

[66] D.C. Creighton, S. Nahavandi, The application of a reinforcement learning agent to a multi-product manufacturing facility, in: Proceedings of the IEEE International Conference on Industrial Technology, Volume 2, 2002: pp. 1229–1234. https://doi.org/10.1109/ICIT.2002.1189350.

[67] H. Cao, H. Xi, S.F. Smith, A reinforcement learning approach to production planning in the fabrication/fulfillment manufacturing process, in: Proceedings - Winter Simulation Conference, IEEE, 2003: pp. 1417–1423. https://doi.org/10.1109/wsc.2003.1261584.

[68] Y.Z. Wei, M.Y. Zhao, Reinforcement learning-based approach to dynamic job-shop scheduling, Zidonghua Xuebao/Acta Automatica Sinica. 31 (5) (2005) pp. 765–771. http://aas.net.cn/fileZDHXB/journal/article/zdhxb/2005/5/PDF/050516.pdf.

[69] Z. Zhang, L. Zheng, M.X. Weng, Dynamic parallel machine scheduling with mean weighted tardiness objective by Q-Learning, International Journal of

Advanced Manufacturing Technology 34 (9–10) (2007) pp. 968–980. https://doi.org/10.1007/s00170-006-0662-8.

[70] Y.C. Fonseca-Reyna, Y. Martínez Jiménez, J.M. Bermúdez Cabrera, B.M. Méndez Hernández, A reinforcement learning approach for scheduling problems, Investigacion Operacional 36 (3) (2015) pp. 225–231. https://www.researchgate.net/publication/282375695_A_reinforcement_learning_approach_for_scheduling_problems (accessed May 3, 2022).

[71] W. Bouazza, Y. Sallez, B. Beldjilali, A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect, IFAC-PapersOnLine. 50 (1) (2017) pp. 15890–15895. https://doi.org/10.1016/j.ifacol.2017.08.2354.

[72] I. Kurinov, G. Orzechowski, P. Hamalainen, A. Mikkola, Automated excavator based on reinforcement learning and multibody system dynamics, IEEE Access 8 (2020) pp. 213998–214006. https://doi.org/10.1109/ACCESS.2020.3040246.

[73] T.M. Moerland, J. Broekens, A. Plaat, C.M. Jonker, A unifying framework for reinforcement learning and planning, (2020). https://arxiv.org/abs/2006.15009.

[74] J. Woo, C. Yu, N. Kim, Deep reinforcement learning-based controller for path following of an unmanned surface vehicle, Ocean Engineering. 183 (2019) pp. 155–166. https://doi.org/10.1016/j.oceaneng.2019.04.099.

[75] H. Dai, E.B. Khalil, Y. Zhang, B. Dilkina, L. Song, Learning combinatorial optimization algorithms over graphs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: pp. 6351–6361. https://dl.acm.org/doi/pdf/10.5555/3295222.3295382 (accessed May 3, 2022)

[76] M. Lu, H.-C. Lam, Critical path scheduling under resource calendar constraints, Journal of Construction Engineering and Management 134 (1) (2008) pp. 25–31. https://doi.org/10.1061/(asce)0733-9364(2008)134:1(25).

[77] M. Lu, H. Li, Resource-activity critical-path method for construction planning, Journal of Construction Engineering and Management 129 (4) (2003) pp. 412–420. https://doi.org/10.1061/(asce)0733-9364(2003)129:4(412).

[78] H. Zhang, H. Li, C.M. Tam, Particle swarm optimization for resource-constrained project scheduling, International Journal of Project Management 24 (1) (2006) pp. 83–92. https://doi.org/10.1016/j.ijproman.2005.06.006.