Simulation of Autonomous Resource Allocation through Deep Reinforcement Learning-based Portfolio-Project Integration

Maryam Soleymani[a], Mahdi Bonyani[a] and Chao Wang[b,*]

[a]Ph.D. Student, Bert S. Turner Department of Construction Management, Louisiana State University, USA
[b]Associate Professor and Graduate Program Advisor, Bert S. Turner Department of Construction Management, Louisiana State University, USA

ARTICLE INFO

Keywords:
Construction Resource Allocation
Construction Simulation
Project Management
Construction Company Portfolio
Deep Reinforcement Learning

ABSTRACT

Resource allocation has always been a critical challenge for construction project planning, and it affects the cost, duration, and quality of the projects. However, current methods mainly focus on a single project and lack integrated planning and optimization across a construction company's multiple projects. This paper describes a simulation of an Autonomous Resource Allocation (ARA) model using Deep Reinforcement Learning (DRL) agents and methods like Double Deep Q-Networks and combined experience replay to develop and test ARA algorithms based on data harvesting from the Internet of Things (IoT) devices. The results show that DRL can successfully perform ARA by capturing the complex interactions among resource allocation features, without needing retraining when situations change. It shows promising future possibilities for construction companies to improve resource utilization and project performance for larger and more complex construction projects.

1. Introduction

In today's highly competitive construction industry, resource management is an important management tool that contributes to improving the performance of construction companies with several projects [1, 2]. Indeed, it is possible to maximize the company's profit by allocating limited resources rationally among competitive objectives, based on the rule of effectiveness [3]. Additionally, the appropriate allocation of resources in construction projects plays a critical role in the planning process as it directly impacts cost, duration, and quality [4]. In addition, the resources available for construction projects are limited, and several activities compete for the same resources [5, 6]. Scheduling by Critical Path Method (CPM) which is a widely accepted project planning technique, does not consider the limitations of these resources [6, 7, 8] and just assumes that activity can begin once its predecessor has finished. As such, in order for a project to succeed, constraints such as schedule deadlines and resources must be met at the same time [9].

However, construction companies do not resource load their schedules frequently since it is very time-consuming. They often create CPM manually using Primavera or Microsoft Project and rarely use intelligent computing and automation. Furthermore, the typical complexities of construction projects are not taken into account by CPM because it is not a dynamic process and has many challenges to be used in practice unless for monthly updates [9, 10, 5]. Therefore, day-to-day update of the project is very difficult and time-intensive since it involves loads of manual steps for a lot of re-sequencing schedules [9, 6]. The first challenge is to allocate resources to different construction processes according to what is needed, when and where it is needed, and what is available [11]. In most classical planning methods, the project plan is developed at the planning phase and it is expected to be executed according to that plan, regardless of any possible changes during the project implementation [4]. However, there are numerous methods used for the optimization of resource allocation and leveling, giving better results compared to traditional methods [2]. In fact, resources must be leveled in the project to avoid difficulties in the construction works due to variations in resource use [12].

Research on optimization of the use of resources by construction companies primarily involves methods of scheduling individual construction projects, rather than a portfolio, taking into account limitations in resource availability. In the meantime, designing an optimum schedule including resource planning for a single building structure (e.g., with a minimum construction cycle) and its implementation according to the schedule do not guarantee a construction company's efficiency. Moreover, guaranteeing a construction company's efficiency requires considering the broader context of portfolio management, addressing resource constraints, adapting to dynamic changes, improving communication and coordination, and making optimization trade-offs. Also, the role of the management staff of a construction company is to maintain a balance between the production capacity of a company and the portfolio of orders [13, 14]. In this way, computerized decision-support systems and optimizing methods can enhance the quality of management decisions [15, 16]. Using a computer-based resource management system, construction companies can keep track of daily updates on their resources, make corrections based on the previous progress of their work on all sites, and update their goals weekly. The purpose is to implement tasks that will help to stick to deadlines for completing construction stages [3, 16].

Autonomous Resource Allocation (ARA) applies efficient and safe resource allocation methods regarding resource constraints and objectives such as time [17, 18]. These methods include coverage resource allocation (CRA)

*Corresponding author
ORCID(s):

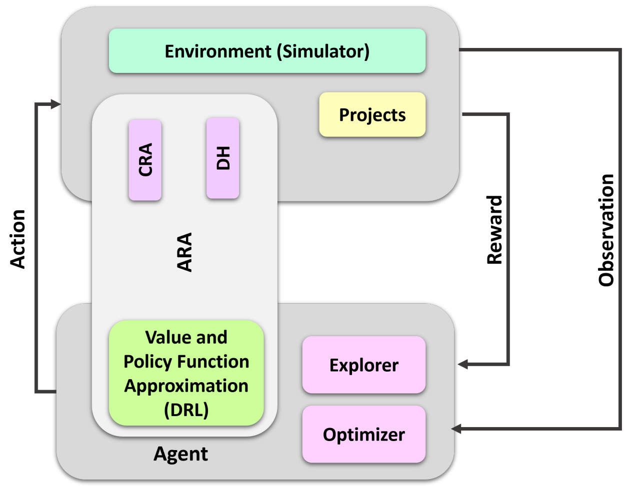

Figure 1: Agent construction company interactions for ARA problem.

[19, 20] for the projects and data harvesting (DH) from Internet of Things (IoT) nodes [21, 22]. Also, CRA and DH methods form the foundation of the ARA system, ensuring effective resource allocation considering both project boundaries and real-time data from IoT devices. Typically, CRA is used to manage resources between the start and end points of the projects, with the goal of covering all points' resource allocation in the project of interest. As far as possible, CRA covers the target project considering obstructions such as constraints in resources [23].

In the construction company, DH scenario involves ARA collecting data from IoT devices distributed throughout the projects' sites, as is shown in Fig. 1, through an alternating line-of-sight (LoS) and non-line-of-sight (NLoS) link [24]. There are many similarities between a DH problem and a CRA problem when presented as an reinforcement learning (RL) problem, since both have very similar constraint sets. The only major difference is the goal function. In previous research, CRA and DH have been studied separately [22, 25, 26, 27, 28]. It is shown that both problems are solvable with deep reinforcement learning (DRL) by putting spatial data of the construction project's company simulation into the DRL agent via layers of convolutional networks [29, 30, 31].

To address the challenges associated with the use of simulation maps as a direct input in the presence of a substantial amount of portfolio information, alternative approaches need to be considered. Large amount of portfolio information makes it problematic to use simulation maps as a direct input because the network's size, training parameters, and training time increase [32]. Our novel solution is to leverage the portfolio and individual project information in an integrated manner, allowing for a more manageable and scalable resource management system. This enables the agent to have a holistic understanding of the available resources across different projects. By providing general portfolio information and incorporating localized details from individual project simulations, the ARA system can strike a balance between efficiency and accuracy in resource allocation. This approach enables the system to overcome the limitations posed by large portfolio datasets while still maintaining the necessary level of detail for effective decision-making.

The DRL provides the possibility of solving both distinctly different problems, namely DH and CRA problems, with the same approach, despite the fact that numerous resource allocation algorithms exist for each problem [33, 34, 35, 36]. As a result, DRL agent learning the management approaches is generalized over a large environment that has variable situations. It does not need retraining when a new resource allocation scenario is encountered as well as recomputing when a new scenario is encountered by changes in situations. Previous studies [37, 38, 39, 40, 41], usually consider only a single scenario when they are determining optimal resource allocation. ARA control tasks are usually nonconvex optimization problems or NP-hard in many instances [42, 43]. So, the DRL concept is suitable in this domain due to its adaptability, computational efficiency, and the complexity of DRL inference. Integer programming [44] and dynamic programming [45] are mathematical models that are often used to find the exact approach to resource allocation [46], which is an NP-hard problem. However, if the practical projects under study are large or complex, these methods may not be computationally feasible or may result in a "combinatorial explosion" problem [47, 48]. These methods include [49, 50, 51, 52] which use priority rules reflecting multiple variables, including the critical index of the activity, the duration, and the minimum late finish time. In spite of this, there are few other heuristic rules that consistently perform better than all others [53]. However, it would be no basis for choosing one rule over another. It is also possible to become trapped within local optima using the general heuristic methods [54]. Tabu search (TS), simulated annealing (SA), and genetic algorithms (GAs) are the three methods of metaheuristics. Repetitive improvements on current solutions are used in SA to achieve better solutions. SA has been applied for resource allocation by [49, 54, 55]. As iterations progress, TS improves the feasible solution so that a local optimum traps it to reach a global one. There are several papers using it to reexamine resource allocation, including [54, 56, 57]. GA has been applied to perform resource allocation based on evolutionary and genetic mechanisms [54, 58, 59, 60]. However, reinforcement learning which is similar to how learning occurs in nature is an area of machine learning. Taking action depends on the outcomes derived from previous actions by an entity called an agent. A positive reinforcement or encouragement of the behavior would reinforce or increase the importance of the action and its actions leading up to it and vice versa (see Fig. 2). The RL approach is based on Markov Decision Processes [61] and differs from supervised learning because it does not require labeled inputs-outputs [62]. In construction projects, the states are unique, and supervised machine learning is impossible since there is no dataset of actions or consequences. There is one type of ARA on which all mentioned approaches focus, and the agent

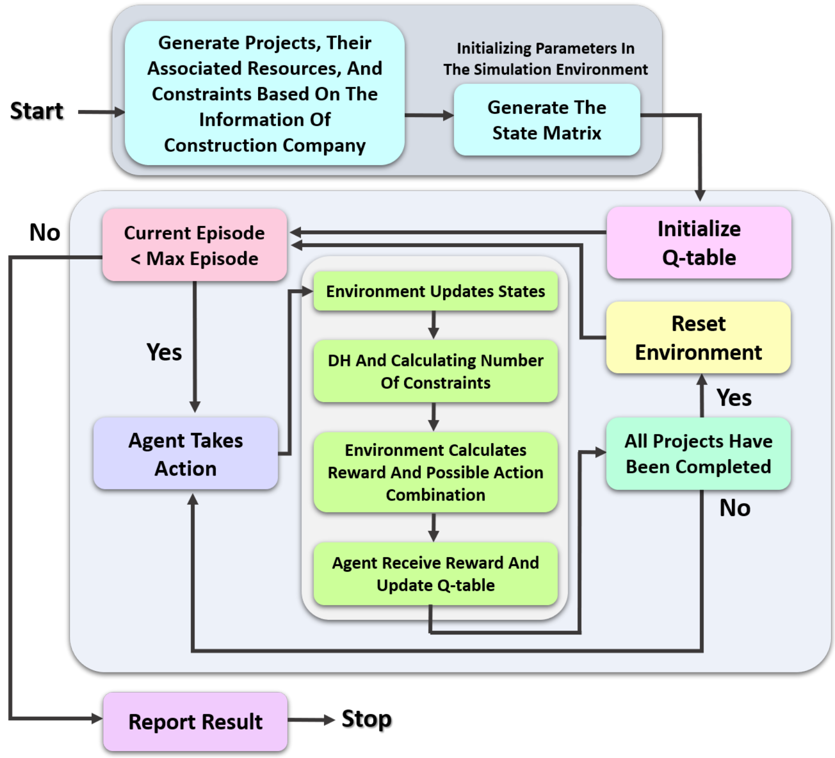

Figure 2: Q-Learning workflow for resource allocation in the construction company.

does not attempt to combine information from portfolio and individual projects. On the other hand, compression of portfolio information reduces computational complexity, but the approach is not RL-based and does not take into account individual projects' information with high precision and hard resource constraints. Overall, to our knowledge, none of the previous studies utilized the methods of parallel processing portfolio and individual projects information that can be applied to various types of resources for ARA. As a result, the main contribution of this paper is a novel approach to ARA in construction companies by leveraging DRL and simulation maps. The integration of CRA and DH enables efficient resource allocation while considering constraints and objectives.

In summary, we suggest the contributions of the proposed method are shown below, and we also summarize all abbreviations used through the context in Table 1.

- We propose a novel approach for processing simulation information portfolios to feed up Deep Reinforcement Learning.
- We propose a novel design reward function to assign efficient resources in different portfolio scenarios.
- Using data harvesting and IoT for information gathering of construction sites.
- The performance and effectiveness of our proposed method was evaluated under the comprehensive set of experimental results in different simulated scenarios.

2. Related Work

ARA in construction project management is a multifaceted challenge, involving the optimization of limited resources such as labor, equipment, and materials. In this section, we delineate the landscape of resource allocation approaches, categorizing them into three categories. Through this categorization, we aim to present a comprehensive overview of existing solutions while paving the way for the introduction of our proposed Deep Reinforcement Learning-based Portfolio-Project Integration Model.

2.1. Metaheuristic Approaches

Metaheuristic approaches have proven instrumental in tackling resource allocation problems by employing innovative search algorithms. Notable techniques include GAs, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO).

GAs, for instance, have demonstrated effectiveness in optimizing resource allocation by minimizing project duration under limited resources [63, 64]. However, challenges arise in selecting appropriate parameter values for specific projects, demanding expert knowledge. Similarly, ACO leverages the collective intelligence of ants to optimize resource allocation in repetitive project activities [65]. Despite their promise, metaheuristic approaches exhibit limitations, such as computational intensity and difficulty in finding globally optimal solutions, especially in complex and nonlinear resource allocation problems.

Table 1
The list of the abbreviation is used in this paper.

Abbreviation	Description
ACO	Ant Colony Optimization
ARA	Autonomous Resource Allocation
CPM	Critical Path Method
CR	Coverage Ratio
CRA	Coverage Resource Allocation
CRAR	Coverage Ratio And Resources
DDQN	Double Deep Q-Network
DH	Data Harvesting
DRL	Deep Reinforcement Learning
ES	Early Start
GA	Genetic Algorithms
IoT	Internet of Things
KPI	Key Performance Indicator
LAS	Look-Ahead Schedule
LAS	Look-Ahead Schedule
LF	Late Finish
LoS	Line-of-Sight
MitC	Mitigation Controller
NLoS	Non-Line-of-Sight
POMDP	Partially Observable Markov Decision Process
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
SA	Simulated Annealing
SNR	Signal-To-Noise Ratio
TS	Tabu Search
TSP	Traveling Salesman Problem

2.2. Mathematical Approaches

Complementing metaheuristic methods, researchers have explored mathematical optimization models and techniques. Huang et al. [66] proposed a GA-based mathematical model for large-scale projects, showcasing improved resource allocation efficiency. Rostami and Bagherpour [67] employed mixed-integer linear programming to address decentralized multi-project scheduling problems. While these approaches offer unique perspectives, they are not without limitations, relying on accurate input data and assumptions.

2.3. Deep Reinforcement Learning Approaches

The application of Deep Reinforcement Learning (DRL) in construction project management has gained significant attention for its ability to handle complex and nonlinear resource allocation problems. In recent studies, DRL has been successfully employed to address various challenges in construction resource allocation. Addressing uncertainties in preventive actions decision-making within infrastructure asset management, Asghari et al. propose a holistic framework that incorporates RL model training [68]. The framework considers deterioration, hazards, and cost fluctuations as uncertainties, while also integrating managerial aspects. Multi-agent RL models are constructed and trained for intervention actions, showcasing improved expected utilities and cost reduction compared to traditional optimization algorithms. Focusing on the adaptive control of labor and material flows in construction projects, Jiang et al. present a model based on Deep Reinforcement Learning. Using a partially observable Markov decision process, the study establishes a mathematical model for resource flow optimization [69]. The proposed DRL-based method demonstrates superior performance compared to conventional optimization methods, offering adaptability to diverse project scenarios. To ensure timely completion of construction projects, Kammouh et al. introduce the Mitigation Controller (MitC), combining nonlinear stochastic optimization techniques and probabilistic Monte Carlo analysis [70]. MitC automates the search for the most effective mitigation strategies on-the-run, considering project manager goal-oriented behavior, contractual performance schemes, and stochastic dependence between construction activities. Soman et al. introduce a novel Look-Ahead Schedule (LAS) generation method using reinforcement learning and linked-data based constraint checking [71]. The proposed method demonstrates the capability to generate conflict-free LAS faster than conventional methods, providing decision support for look-ahead planning meetings. By integrating linked-data based constraint checking within the reward function, the study extends existing knowledge in the construction informatics domain.

The rationale for our study is grounded in the observed limitations of existing research in the field of resource allocation within construction companies. The existing research has predominantly focused on scheduling individual construction projects while they considering resource availability constraints. However, our study seeks to extend beyond this approach by emphasizing the importance of portfolio-level resource allocation. To overcome the limitations identified in existing approaches, we propose the integration of Deep Reinforcement Learning techniques. RL leverages trial and error, learning, and adaptive decision-making, offering a dynamic and robust solution for resource allocation. Unlike traditional methods, RL can handle complex and nonlinear resource allocation problems, providing adaptive strategies by continuously learning and adapting to changing project conditions.

In the following section (Section 6), we present our Deep Reinforcement Learning-based Portfolio-Project Integration Model, aiming to enhance the effectiveness and efficiency of resource allocation in construction companies. This model addresses the gaps identified in traditional approaches, offering a promising avenue for the advancement of autonomous resource allocation in construction project management.

3. Problem Formulation for Resource Allocation Modeling

3.1. Unification of Construction Company Simulation and ARA Model

In the following, it is shown how two parts of the problem are separated: the construction portfolio information and the individual projects' target, which can make the problem description universal.

Considering a square grid of size $C \times C \in \mathbb{N}^2$ with a portfolio of size m, where $\mathbb{N}$ is the set of natural numbers. Individual projects' start/finish boundaries, constraints of resources, and other obstacles such as delays in DH make up the construction company simulation. It can be described by a tensor $\mathbf{C} \in \mathbb{R}^{C\times C\times 3}$, in which $\mathbb{R} = \{0,1\}$ and start/finishing individual projects make up project-layer 1, constraints of resources and obstacles make up project-layer 2, and obstacles alone make up project-layer 3.

Constant individual projects k is maintained by the ARA through this construction company simulation as it occupies individual projects. Therefore, its project can be defined in terms of $\mathbf{p}(t) \in \mathbb{N}^2$. A collision-avoidance strategy and constraints on resources restrict the selection of individual projects by the ARA. The ARA must start and end its tasks without exceeding its minimum delay time determined by its initial resource level in any project that belongs to the start and finish projects. Every time a step in the action is completed and set to $r_0 \in \mathbb{N}$ at time $t = 0$, the resource level of the project $r(t)$ decrements by 1.

The following subsection delves into the specific objectives and definitions related to the target project and the DH component. Moreover, we focus on minimizing delay time and ensuring coverage within the information field based on the target project, while the DH component involves data harvesting from IoT devices scattered across the construction portfolio. By unifying the construction project-layer description, both problems can be addressed using deep reinforcement learning based on a neural network.

3.2. Target Project and Objectives Definitions

1) CRA: When allocating resources for coverage, the objective is to minimize delay time or project time, such that falls inside the information of a construction sensor installed on the individual project. $\mathrm{TP}(t) \in \mathbb{R}^{C\times C}$ can describe the target project, where each resource describes whether a project needs to be covered. Each project can be classified as belonging to the current field of information by using $\mathbf{V}(t) \in \mathbb{R}^{C\times C}$ to indicate whether it belongs to it. This work uses a simulation field of information with a $5x5$ size adjacent to the current position of the ARA. The $\mathrm{V}(t)$ calculation also takes into account obstructions of Line-of-Sight. This prevents the ARA from looking around the corner.

As a result, the target project develops in accordance with Eq. 1,

$$TP(t+1) = TP(t) \wedge \neg V(t) \tag{1}$$

There are two types of logical operators $\wedge$ and $\neg$ which represent logical "and" and "negations". Starting and finishing points of the projects as well as constraints of resources can be covered by targets, whether stopped projects are in the construction company simulation or not. We try to complete as much of the target project within the minimum time delay as possible.

2) DH: In contrast, wireless data harvesting is utilized to obtain information on residual resources from $M \in \mathbb{N}$ terminal IoT devices scattered across the construction portfolio at simulation, with $m \in [1, M]$ situation given by $\mathbf{u}_m \in \mathbb{N}^2$. The ARA must collect a certain amount of data $D_m(t) \in I$ from each device. According to the usual log-resource loss model with Gaussian shadow fading, over the selected device m, their data throughout $C_m(t)$ is determined, whether the ARA includes stopped projects or not. One device at a time communicates with the ARA, which is selected regarding the most available data resources and the highest resource requirements. In [21], the performance of the link and multiple access protocols are described in more detail. In each device, data evolves in accordance with Eq. 2,

$$D_m(t+1) = D_m(t) - C_m(t) \tag{2}$$

Every project can have a device. Data harvesting must collect all the data from the devices within the shortest possible time frame and with minimum delay in order to collect as much data as possible.

3) Unifying Construction Project-Layer Description: $\mathbf{D}(t)$
$\in I^{C\times C}$ is the target project layer which can be used to describe both problems. According to (1), CRA provides the target project layer through $\mathrm{T}(t)$. Target project-layers in DH show how much data is available in each project in which one of the devices is located, so that project u_m has value $D_m(t)$ and is evolving according to (2). The value of a project is 0 if a device does not exist in the project or 1 if all the device data have been collected. Deep reinforcement learning based on a neural network with the same structure can solve both problems since their state representations are similar.

3.3. IoT Modeling

Several key models of the ARA-based resource allocation are presented in the following. In order to implement the RL approach, the construction site's dynamic needs to be simplified to a certain extent. We make explicit our assumptions whenever possible.

IoT cameras, integrated with advanced sensors and connectivity capabilities, offer a comprehensive monitoring system that captures valuable data on construction resources. These intelligent cameras leverage the power of computer vision, artificial intelligence, and cloud computing, enabling remote access to real-time video feeds and valuable insights [72]. By deploying IoT cameras strategically across construction sites, project managers gain a high-level overview of resource utilization, enabling them to make informed decisions and optimize resource allocation.

One of the key advantages of IoT cameras in construction resource monitoring is their ability to provide a continuous and uninterrupted stream of data. Construction sites are dynamic environments with constantly changing conditions and activities [73]. IoT cameras capture and transmit visual information, allowing stakeholders to monitor automatic construction progress, identify bottlenecks, and assess the availability of essential resources such as materials, machinery, and personnel. This real-time monitoring facilitates proactive decision-making, reducing downtime and delays that may arise from resource shortages or unexpected events.

Our assumptions are explicit whenever suitable. Our approach also introduces the concept of communication time slots in addition to decision time slots, since communication systems are typically operated on a longer timescale than ARA's decision resource management system. A number of communication time slots $\gamma \in \mathbb{N}$ are divided into each decision time slot $t \in [0, T]$. After that, $n \in [0, N]$ with $N = \gamma T$ becomes the communication time index. There are $\mathrm{A}_n = \mathrm{g}_t/\gamma$ seconds in one communication time slot n. As a result, we choose γ a sufficiently large number of communication time slots so that the ARA decision can be made linearly between $\mathbf{p}_l(t)$ and $\mathbf{p}_l(t+1)$, within each communication time slot. $d \in \mathrm{D}$ is the d-th loT device. $G_d(t) \in \mathbb{R}^+$represents the amount of data that must be collected by each sensor over the course of the decision $t \in [0, T]$. An initial volume is set for the device data volume at the start of a decision $G_d(t=0) = G_{d,\mathrm{init}}$. Over the course of a decision, each IoT node's data volume changes based on the communication time index n, which is expressed as $G_d(n)$ in conjunction with $n \in [0, N], N = \gamma T$. There is a log-distance path loss and shadow fading between ARA and D IoT devices during communication over LoS/NLoS point-to-point channels. At time n, the d-th device can send the maximum information rate by Eq. 3,

$$IR_{l,d}^{\max}(n) = \log_2\left(1 + \mathrm{SNR}_{l,d}(n)\right) \tag{3}$$

Considering the amount of information accessible at the d-th device $G_d(n)$, the effective information rate is given as shown in Eq. 4,

$$IR_{l,d}(n) = \begin{cases} IR_{l,d}^{\max}(n), & G_d(n) \geq \mathrm{R}_n IR_{l,d}^{\max}(n) \\ G_d(n)/\delta_n, & \text{otherwise.} \end{cases} \tag{4}$$

The Signal-To-Noise Ratio (SNR) is modeled as a Gaussian random variable in Eq. 5.

$$\mathrm{SNR}_{l,d}(n) = \frac{TP_{l,d}}{\sigma^2} \cdot g_{l,d}(n)^{-\alpha_\varepsilon} \cdot 10^{\eta_\theta/10} \tag{5}$$

While $TP_{l,d}, \sigma^2, g_{l,d}$, and α_e are Transmit power, receiver Gaussian noise power, camera-device distance, and path loss exponent, respectively. It is worth noting that construction sites present various challenges that impede the smooth propagation of signals, leading to a significant influence of propagation parameters on the conditions of LoS as well as NLoS. The equation represented as Eq. 5 denotes the SNR averaged over small-scale fading. It is essential to highlight that our newly proposed approach is model-free, eliminating the need for any specific channel transmission model. Although a more precise and intricate model could potentially be used in conjunction with our approach, Eq. 5 already encompasses the crucial aspects for data collection of the urban channel, namely the correlation between SNR and the variables $g_{l,d}$ and the conditions $e \in \{LoS, NLoS\}$.

Overall, the integration of IoT cameras and the underlying equations plays a crucial role in addressing resource allocation challenges through the RL approach. IoT cameras, equipped with advanced sensors and connectivity capabilities, establish a comprehensive monitoring system that captures real-time data on construction resources. The continuous stream of data from IoT cameras enables proactive decision-making and reduces downtime caused by resource shortages or unforeseen events. In the context of the presented equations, the communication time slots and decision time slots facilitate efficient communication between the ARA system and IoT cameras. The equations, such as Eq. 4 and Eq. 5, establish the foundation for modeling the information rate and signal-to-noise ratio, enabling effective data collection and transmission. Through simulation and iterative RL processes, the ARA system can learn and adapt its resource allocation strategies based on the real-time data obtained from IoT cameras. This integrated IoT-camera-equation framework provides a valuable tool for addressing resource allocation challenges in construction sites, ultimately improving project performance and resource efficiency.

3.4. RL Formulation

The following approach can be applied to both distinct allocations of resource problems, despite the fact that there are a variety of methods for solving them separately. Typically, individual projects are connected by resources' limitations into a graph, and each project is covered by an allocation of resources in classical CRA approaches. As a result, CRA becomes an instance of the traveling salesman problem (TSP), which can be handled by standard approaches, such as [74], but at the expense of exponentially increasing time complexity. IoT devices can serve as nodes in a graph and distances between them as edge costs in a TSP. However, the conversion doesn't take into account communications between the device and the ARA while traveling to or from the device. A sequential visit of all devices isn't usually the optimal behavior in DH problems. Instead, information can already be obtained more efficiently by building a LoS link farther away, or the ARA may have to hover near a device for an extended period to collect information in large quantities. It is not trivial to model and solve these constraints with classical techniques, coupled with stochastic communication channel models and multiple access protocol options. It is often not possible to cover or collect all the data for both problems due to the constraints of ARA and resources. With the DRL methodology, we can directly combine all of the resource allocation goals and constraints without approximations.

The following subsection introduces the concept of Partially Observable Markov Decision Processes (POMDPs) and explains how they can be utilized to model resource allocation problems. It provides a detailed understanding of the state space, action space, and reward functions within the POMDP framework, allowing for the application of RL techniques to effectively address the resource allocation challenges.

4. Resource Allocation Simulation through Partially Observable Markov Decision Processes

According to challenges of resource allocation within dynamic and partially observable environments, the application of POMDPs emerges as a powerful framework. POMDPs, as highlighted by Kaelbling et al. [75] and Sutton and Barto [76], address challenges arising from partial observability, uncertainty, and sequential decision-making. In our context, deploying POMDPs is motivated by the inherent complexities of construction company portfolio simulations, where the availability of information is limited, and decision-making must account for dynamic project states [77].

To grasp the essence of our POMDP-based approach, it's essential to dissect the components encapsulated within the POMDP tuple $(\mathcal{S}, \mathcal{A}, P, R, \Omega, \mathcal{O}, \gamma)$:

1. $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, respectively. In our scenario, $\mathcal{S}$ encapsulates information about the construction company's portfolio simulation, target projects, situational details, and time delays.
2. The transition probability function $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ captures the likelihood of transitioning between states given an action, reflecting the dynamism of the construction environment.
3. The reward function $R : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ encapsulates various aspects, including positive rewards for information collection, safety considerations, penalties for decision delays, and resource allocation efficiency.
4. Ω and $\mathcal{O} : \mathcal{S} \mapsto \Omega$ define the observation space and observation function, respectively. These elements introduce the concept of partial observability, acknowledging the limitations in perceiving the entire construction environment.
5. The discount factor $\gamma \in [0, 1]$ differentiates between the valuation of long-term and short-term resource allocation rewards, contributing to the temporal aspect of decision-making.

A pivotal aspect of our methodology is the introduction of a unified representation of the resource state within the state space $\mathcal{S}$. By unifying these diverse elements, we facilitate a holistic view for autonomous resource allocation problems, streamlining decision-making processes. The use of POMDPs in our resource allocation simulation offers practical advantages. It enables the ARA to make-decision through a complex construction landscape with incomplete information, dynamically adapting to evolving project states. This adaptability is crucial in scenarios where the ARA must contend with uncertainties, project constraints, and the need for efficient resource allocation. To unify the ARA problems, a unified representation of the resource state is introduced to integrate into a cohesive and manageable structure. The state space $\mathcal{S}$ is defined as:

$$\mathcal{S} = \underbrace{\mathbb{R}^{C \times C \times 3}}_{\substack{\text{Construction Company} \\ \text{Portfolio Simulation}}} \times \underbrace{I^{M \times M}}_{\substack{\text{Target} \\ \text{Project}}} \times \underbrace{\mathbb{N}^2}_{\text{Situation}} \times \underbrace{\mathbb{N}}_{\substack{\text{Time} \\ \text{Delay}}} \tag{6}$$

Let's delve into the components that constitute this unified representation:

Construction Company Portfolio Simulation ($\mathbb{R}^{C \times C \times 3}$): The construction company's portfolio simulation is encapsulated in a three-dimensional matrix, incorporating essential details such as project start and end points, resource constraints, and the status of ongoing projects. This matrix, denoted as $\mathbf{C}$, provides a comprehensive snapshot of the construction landscape.

To simulate the company portfolio as shown example in Table 7 in a 3D matrix, we need to map the table columns and rows into the dimensions of the matrix. Since the table represents different simulation scenarios with various parameters, this paper considers each dimension of the matrix to represent one aspect of the scenario. Let's define the dimensions of the matrix as follows for Table 7 as an example:

Dimension 1 : This dimension can represent the number of projects, where 0 corresponds to 5 projects and 1 corresponds to 10 projects. Dimension 2 : This dimension can represent the number of tasks per project, where 0 corresponds to 5 tasks, 1 corresponds to 10 tasks, 2 corresponds to 20 tasks, and 3 corresponds to 40 tasks. Dimension 3: This dimension can represent the number of resource types, where 0 corresponds to 10 resource types, 1 corresponds to 20 resource types, and 2 corresponds to 40 resource types.

A pseudo-code representation of the algorithm to create and populate the 3D matrix based on the provided table is illustrated in Algorithm 1.

Target Project Information ($I^{M \times M}$): Within the unified state space, we allocate space for target project information represented by the matrix $I^{M \times M}$. This matrix captures critical details about individual projects, including their existence, discovery status, and potential resource availability.

Situational Context ($\mathbb{N}^2$): Situational context is incorporated as a two-dimensional vector, denoted as $\mathbb{N}^2$. This vector encapsulates relevant information about the construction environment, contributing to the contextual awareness of the ARA.

Time Delay ($\mathbb{N}$): Acknowledging the temporal dimension, we include a scalar representing time delay within the unified state space. This temporal information is crucial for guiding the ARA in making decisions considering the evolving nature of the construction projects.

Algorithm 1 Simulation process of construction company portfolio

Require: Initialize the 3D matrix with zeros.
Ensure: values = Create the random values for the number of project, task, and resource.

```
Output = matrix
for i ← 1 to C Grid Size do
    proj_idx ← floor(i/C)        ▷ index of the project
    task_idx ← floor(i/C)        ▷ index of the tasks per project.
    scenario_values ← values[i]
    for j ← 0 to 3 Constraint type do
        constraint_value ← scenario_values[j]
        matrix[proj_idx][task_idx][j]← constraint_value
    end for
end for
```

Consider the unified representation components $s(t) \in \mathcal{S}$ are

$$s(t) = (\mathbf{C}, \mathbf{D}(t), \mathbf{p}(t), r(t)) \tag{7}$$

Let's break down the interpretation of each component:

- $\mathbf{C}$ denotes the construction company portfolio simulation,capturing the holistic view of the construction company's portfolio simulation. It includes details about project timelines, resource constraints, and the status of stopped projects.

- $\mathbf{D}(t)$ represents project indicating whether there is any more data at the device on construction portfolio simulation or whether any more individual projects need to be discovered at time t. It accounts for the existence of data, the need for project discovery, and other project-specific details.

- $\mathbf{p}(t)$ represents the ARA state at time t and captures the ongoing decision-making process.

- $r(t)$ represents the ARA remaining decision of individual project at time t. It is a critical parameter guiding the ARA in managing time-sensitive resource allocation decisions.

The action $a(t) \in \mathcal{A}$ of the ARA at time t is listed as one of the alternative actions

$$\mathcal{A} = \{ \text{ next, previous, assign, ignore, hold } \}$$

The reward function plays a pivotal role in shaping the decision-making process of the ARA. It is designed to provide a quantitative measure of the desirability of different states and actions, guiding the ARA towards optimal resource allocation strategies. Tuning parameters within the reward function allows for flexibility in adapting to different construction scenarios and project priorities. The reward function's adaptability is crucial for ensuring that the ARA can dynamically adjust its decision-making strategies based on the evolving nature of the construction projects. There are four elements that make up the generalized reward function $R(s(t), a(t), s(t+1))$:

- r_c (positive): used to cover the reward derived from the collected information or the number of new target projects covered, comparing $s(t+1)$ and $s(t)$.

- r_{sc} (negative) When an ARA must avoid colliding with a project or constraints of resources, a safety controller penalty will be imposed.

- r_{dec} (negative) represents the penalty When the ARA does not complete the tasks or make decisions. This component encourages the ARA to prioritize tasks efficiently, penalizing delays or failure to complete assigned tasks within the defined time frame.

- $r_{\text{delay (negative) penalty for residual resource}}$ penalizes the delay in resource utilization due to the safe initiation (time delay gets zero) of individual projects. It assesses the ARA's ability to manage time effectively. It rewards decisions that contribute to minimizing delays in task execution, ensuring timely project completion.

These reward elements capture various aspects of the ARA's decision-making process and the associated trade-offs including information coverage, safety considerations, task completion penalties, and time delay management. By considering these rewards, the RL framework can guide the ARA in making optimal resource allocation decisions while considering safety, completion of tasks, and minimizing delays. The POMDP formulation, as outlined above, guides the ARA in making optimal decisions under fluctuating project statuses, unforeseen delays, and changing resource requirements challenging conditions.

The next subsection provides a detailed explanation of the construction company portfolio simulation processing. These processed simulation maps play a crucial role in the subsequent stages of the methodology, aiding in improving the agent's performance and facilitating the interpretation of the resource state.

4.1. Construction Company Portfolio Simulation Processing

It is necessary to use two simulation map processing steps in order to facilitate the interpretation of the large resource state presented in Eq. 6. Also, for the purpose of improving the agent's performance, the simulation mapping should be centered around its position. By using this approach, the resource state representation size is further increased, which is an advantage. Moreover, the centered simulation map is presented as two inputs: detailed individual project information, showing the agent's unreliable surroundings, and squeezed portfolio information, showing the whole portfolio. This is how the three functions are mathematically described, which compose the main contribution

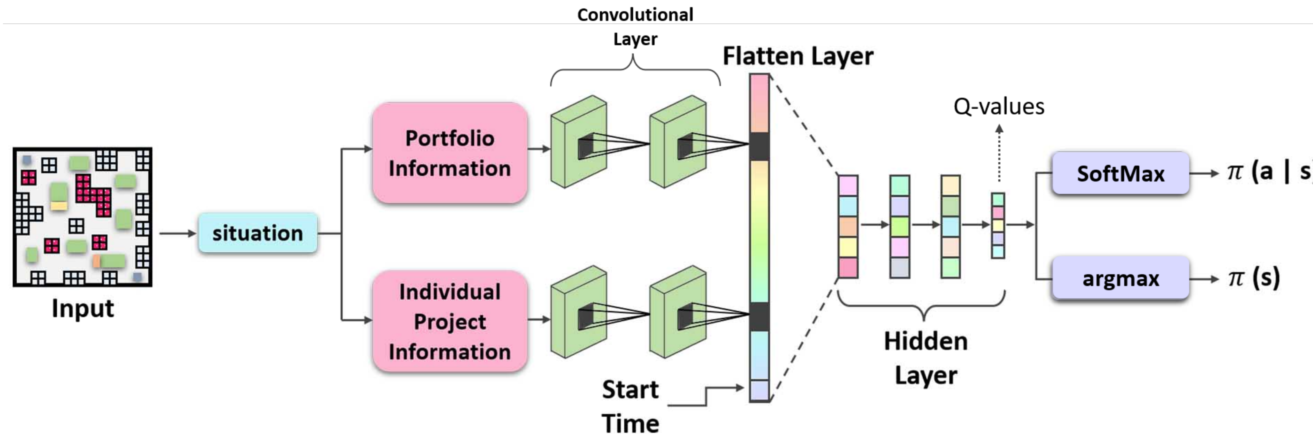

Figure 3: Portfolio and Individual Project Information Simulation Mapping with DDQN architecture.

of this work. An illustration of the data pipeline can be found in Fig. 3.

Given a tensor $\mathbf{A} \in I^{C \times C \times n}$ describing the layers of the construction company portfolio, a centered tensor $\mathrm{B} \in I^{C_m \times C_m \times n}$ with $C_m = 2C - 1$ is defined as follows:

$$\mathbf{B} = f_{\text{center}}\left(\mathbf{A}, \mathbf{p}, \mathbf{x}_{\text{pad}}\right) \tag{8}$$

According to the following definition of the centering function:

$$f_{\text{center}} : I^{C \times C \times n} \times \mathbb{N}^2 \times I^n \mapsto I^{C_m \times C_m \times n} \tag{9}$$

The components of $\mathbf{B}$ according to the components of $\mathbf{A}$ are defined as:

$$\mathbf{b}_{i,j} = \begin{cases} \mathbf{a}_{i+p_0-C+1, j+p_1-C+1}, & C \leq i + p_0 + 1 < 2C \\ & \wedge C \leq j + p_1 + 1 < 2C \\ \mathrm{x}_{\text{pad}}, & \text{otherwise} \end{cases} \tag{10}$$

The padding value $\mathbf{x}_{\text{pad}}$ is effectively applied to the project layers of the construction company portfolio of $\mathbf{A}$. There is a vector value of dimension I^n in $\mathbf{a}_{i,j}, \mathbf{b}_{i,j}$, and a vector value of dimension $\mathbf{x}_{\text{pad}}$. There are two problems, namely constraints and stopped projects. To solve them, the construction layers are padded with $[0, 1, 1, 0]^{\mathrm{T}}$. Centering is qualitatively explained in [21] and illustrated with an example. Portfolio and individual project information simulation mapping cannot be performed without using the tensor $\mathbf{B} \in I^{C_m \times C_m \times n}$, which is the result of the simulation map centering function in which portfolio simulation map scaling is called *port* and project simulation map size is called *proj*. At the initial step, the simulation map of an individual project must be created according to:

$$\mathbf{X} = f_{\text{project}}(\mathbf{B}, proj) \tag{11}$$

by using the individual project simulation map function defined by:

$$f_{\text{project}} : I^{C_m \times C_m \times n} \times \mathbb{N} \mapsto I^{\text{proj} \times proj \times n} \tag{12}$$

When X is compared to B, the following elements are presented:

$$\mathbf{x}_{i,j} = \mathbf{b}_{i+C-\lfloor \frac{l}{2} \rfloor, j+C-\lceil \frac{l}{2} \rceil} \tag{13}$$

As a result of this operation, a size $l \times l$ central crop has been obtained.

In order to create the portfolio simulation map, the following criteria are used:

$$\mathbf{Y} = f_{\text{portfolio}}(\mathbf{B}, port) \tag{14}$$

$$f_{\text{portfolio}} : I^{C_M \times C_M \times n} \times \mathbb{N} \mapsto \mathbb{R} \left| \frac{C_m}{\text{port}} \right| \times \left| \frac{C_m}{\text{port}} \right| \times n \tag{15}$$

$\mathbf{Y}$ is composed of the following elements when compared to $\mathbf{B}$:

$$\mathbf{y}_{i,j} = \frac{1}{port^2} \sum_{u=0}^{port-1} \sum_{v=0}^{\text{port-1}} \mathbf{b}_{port, i+u, port, j+v} \tag{16}$$

which is an operation equal to the average pooling.

By using l and g, respectively, we can parameterize the functions f_{project} and $f_{\text{portfolio}}$. Project simulation maps are larger when l is increased, whereas the average pooling of individual projects is larger when g is increased, resulting in portfolio simulation maps being smaller.

These simulation processing steps allow for a more manageable representation of the construction company portfolio and individual projects. The resulting simulation maps provide focused and compact information that is conducive to effective decision-making by the agent. The data pipeline, as illustrated in Figure 3, showcases the flow of information from the construction company portfolio to the creation of individual project and portfolio simulation maps, highlighting the importance of these processing steps in the methodology. In 'Site32', the model architecture consists of two convolutional branches, portfolio and individual project.

For portfolio, the initial input size at layer 1 is 21x21x4, representing a 4-channel input image with a spatial dimension of 21x21. This input undergoes convolutional operations, resulting in an output size of 17x17x16 in layer2. The subsequent layer, layer2, further transforms this feature map to a size of 13x13x16. Simultaneously, individual project starts with an initial input size of 17x17x4, which is processed through convolutional layers to produce an output size of 13x13x16 in conv layer 2. The two branches are then concatenated, creating a composite feature map that is fed into a three fully connected layer with a size of 256 neurons. In 'Site50', there are adjustments in the input sizes for both branches. For portfolio, the initial input size at layer 1 is now 19x19x4, leading to a conv layer 2 output size of 15x15x16. Similarly, individual project maintains the same input size configuration as in State 1 (17x17x4), resulting in an identical conv layer 2 output size of 13x13x16. The concatenated feature maps from both branches are then passed through three fully connected layers with a size of 256 neurons, maintaining consistency with 'Site32'.

The POMDP formulation provides the theoretical foundation for modeling the resource allocation problems, while the simulation processing steps enhance the practical implementation of the methodology. By combining the POMDP formulation with simulation processing techniques, the methodology achieves a comprehensive and practical approach to solving resource allocation problems. The simulation maps generated through the processing steps serve as inputs to the POMDP framework, allowing for more accurate and focused decision-making by the agent. As a result, these components provide a powerful framework for addressing resource allocation problems, enabling effective decision-making in complex and dynamic construction company environments.

The following subsection introduces the observation space Ω, consisting of individual project and portfolio observations, and outlines the mapping functions to generate these observations. The use of partial observability and compression techniques improves the efficiency and scalability of the methodology, enabling the agent to effectively handle resource allocation problems in a resource-constrained environment.

4.2. Observation Construction Company Portfolio Simulation

The observation space, denoted by Ω, is defined as the product of two components: Ω_l and Ω_g. Ω_l represents the observation of individual projects, while Ω_g represents the observation of the portfolio. The individual project observation is a fundamental element of the observation space, capturing detailed information about each project within the construction company portfolio like project progress status and resource requirements and constraints. Also, the portfolio observation provides a holistic view of the entire construction company portfolio, allowing the ARA to understand the broader context in which resource allocation decisions are made such as aggregate resource utilization across all projects and global resource constraints and availability.

Ω which is the observation company portfolio simulation for the agent, is given as follows:

$$\Omega = \Omega_l \times \mathbb{N} \times \Omega_g$$

When $\Omega_l = \mathbb{R}^{\text{proj} \times \text{proj} \times 3} \times I^{\text{proj} \times proj}$ and $\Omega_g = I^{\left|\frac{C_m}{\text{port}}\right| \times \left|\frac{C_m}{\text{port}}\right| \times 3} \times I^{\left\lfloor \frac{C_m}{\text{port}} |\times| \frac{C_m}{\text{port}} \right\rfloor}$ are applied to the project and portfolio simulation map, respectively, and the project-layers are compressed using the average pooling, the company portfolio layers become real instead of boolean. By using the tuple, we define the observations $o(t) \in \Omega$:

$$o(t) = \left(\mathbf{C}_{proj}(t), \mathbf{D}_{\text{proj}}(t), \mathbf{C}_{port}(t), \mathbf{D}_{port}(t), r(t)\right) \quad (17)$$

A project and portfolio observation of the company portfolio simulation is represented by $\mathbf{C}_l(t)$ and $\mathbf{C}_g(t)$ in the observation. A project observation is $\mathrm{D}_l(t)$, and a portfolio observation is $\mathrm{D}_a(t)$. There is a similar time delay $r(t)$ for the ARA in the resource state. Because the ARA moves along a time-dependent path, both the project and portfolio simulation observations are time-dependent. The mapping function $\mathcal{O} : \mathcal{S} \mapsto \Omega$ describes how the resource state is transformed into observation resources as shown in Eq. 18. The function defines the observation components based on the project and portfolio simulation maps. It utilizes the functions f_{project} and $f_{\text{portfolio}}$ to create the project and portfolio observations, respectively. These functions employ the centered simulation maps generated in the previous subsection and apply the necessary compression techniques to obtain the desired observation representations.

$$\begin{aligned}
\mathbf{C}_{\text{proj}}(t) &= f_{\text{project}}\left(f_{\text{center}}\left(\mathbf{C}, \mathbf{p}(t), [0,1,1]^{\mathrm{T}}\right), \text{ proj }\right) \\
\mathbf{D}_{\text{proj}}(t) &= f_{\text{project}}\left(f_{\text{center}}(\mathbf{D}(t), \mathbf{p}(t), 0), proj\right) \\
\mathbf{C}_{\text{port}}(t) &= f_{\text{portfolio}}\left(f_{\text{center}}\left(\mathbf{C}, \mathbf{p}(t), [0,1,1]^{\mathrm{T}}\right), \text{ port }\right) \\
\mathbf{D}_{\text{port}}(t) &= f_{\text{portfolio}}\left(f_{\text{center}}(\mathbf{D}(t), \mathbf{p}(t), 0), \text{ port }\right)
\end{aligned} \quad (18)$$

In this case, the issue is intentionally transformed into a partially visible Markov decision processes resource Ω by supplying it to the agent instead of the resource state $\mathcal{S}$. Project simulation maps are restricted in size, and portfolio simulation maps are averaged. Hence, partial observability results show that partial reliability does not exacerbate the problem unsolvable for memoryless agents, as well as that the neural network becomes significantly smaller after compression, resulting in substantially less training time.

The simulation processing steps presented in Subsection 4.1 serve as a preprocessing stage to create simulation maps, which are then utilized in the construction of observation resources discussed in Subsection 4.2. The simulation maps, representing the construction company portfolio and individual projects, are centered and compressed to provide focused and concise information. These maps, along with the time delay information, form the observation resources that enable the agent to make informed decisions.

By combining the simulation processing techniques with the construction of observation resources, the methodology achieves a comprehensive approach to solving resource allocation problems. The simulation maps provide a realistic and interpretable representation of the construction environment, while the observation resources enable the agent to effectively navigate and allocate resources within this environment. As a result, these components form a cohesive framework for addressing resource allocation problems. The simulation processing steps provide focused information, while the observation resources offer a partially observable view of the resource state, enhancing the efficiency and effectiveness of the decision-making process.

These techniques and parameters contribute to the efficient learning and decision-making process of the agent in solving the resource allocation problems within the construction company simulation.

4.3. Double Deep Reinforcement Learning - Neural Network

A reinforcement learning approach can be used to solve the POMDP outlined above, specifically double deep Q-networks (DDQNs) [43]. According to DDQNs, each pair of state-action values is approximated as follows:

$$Q^{\pi}(s(t), a(t)) = \mathbb{E}_{\pi}\left[\sum_{k=t}^{T} \gamma^{k-t} I(s(k), a(k), s(k+1))\right] \quad (19)$$

An agent approximates the state-action values using a discount factor γ and an expectation over the policy π. A replay memory $\mathcal{D}$ stores $\left(s, a, r, s'\right)$ the experiences $(s(t), a(t), i(t), s(t+1))$ that the agent collects as it explores the construction company simulation, omitting temporal information to converge to the optimal Q-value. θ and $\bar{\theta}$ are used to parametrize two Q-networks. θ updates the first Q-network by reducing and minimizing the squared difference between the current Q-value estimate and the target value as described in Eq. 20. The target value is calculated based on the reward $r(s, a)$ and the maximum Q-value of the next state.

$$L(\theta) = \mathbb{E}_{s,a,s' \sim \mathcal{D}}\left[\left(Q_{\theta}(s, a) - Y\left(s, a, s'\right)\right)^2\right] \quad (20)$$

The replay memories are based on experiences. The target value is as follows:

$$Y\left(s, a, s'\right) = r(s, a) + \gamma Q_{\bar{\theta}}\left(s', \underset{a'}{\operatorname{argmax}} Q_{\theta}\left(s', a'\right)\right) \quad (21)$$

A soft update parameter $\tau \in (0, 1]$ is applied to the parameters of the second Q-network as $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$. Moreover, The second Q-network parameters gradually approach the parameters of the first Q-network. As a means of addressing training sensitivity to replay memory size, Zhang and Sutton [78] propose combined experience replay. This technique helps stabilize the training process by sampling experiences from multiple replay memories.

Fig. 3 depicts how both Q-networks are designed using neural networks. Project and portfolio observation components are generated by stacking and centering the target resource assignment and construction company simulation around the ARA. The constructed tensors are put into two branches of convolutional layers which then flatten and concatenate with the remaining time to the start point of individual projects before passing through three hidden layers using Rectified Linear Unit (ReLU) activation functions [79]. Employing a SoftMax function for exploration or argmax function for exploitation, a layer without an activation function represents the Q-values directly. To determine scalability, the flatten layer should be of a size that provides adequate scalability. It is calculated by the following equation:

$$N = n_c\left(\left(\text{proj} - n_k\left\lfloor\frac{S_k}{2}\right\rfloor\right)^2 + \left(\lfloor\frac{C_m}{\text{port}}\right\rfloor - n_k\left\lfloor\frac{S_k}{2}\right\rfloor\right)^2\right) + 1 \quad (22)$$

with n_c representing the number of kernels, n_k representing the number of convolutional layers, and s_k representing the number of kernel sizes. A portfolio-project construction company simulation processing scenario with $port = 1$ scaling parameter and $proj = 0$ size parameter means that there will be no down-sampling and no additional project simulation map. In Table 2, the used parameters for evaluation were listed. Also, determining appropriate parameters is a crucial step to ensure the model's effective performance. Parameters like learning rate, batch size, the number of convolution and fully connected layers, Number of kernels, Convolution kernel width, and the number of hidden units are considered as hyperparameters, were determined through a grid search.

As a result, the combination of DDQNs and the neural network architecture presents a promising approach for tackling the resource allocation challenges in the construction industry. By leveraging reinforcement learning techniques and neural network modeling, the ARA can learn effective strategies for allocating resources in dynamic and partially observable environments, improving the overall efficiency and performance of construction projects.

4.4. Simulation Setup

The ARA decides in two different construction company simulations. In Fig. 4 (a and c), 'Site32' has a grid of 32 by 32 projects with two starting and finishing points on the top left and bottom right corners. Stopped projects are also presented along with regular construction site patterns. There are a start point and a finish point around the target project in the 'Site50' scenario Fig. 4 (b and d). Stopped projects are more prominent at the bottom of the simulation map since they are generally larger and more spaced out. As the 'Site50' simulation map shows, it includes roughly larger projects. The legend for the plots provided by Table 3 shows the project sizes for the scenarios.

Table 2
Hyperparameters for two different construction company simulations.

Description	32*32	50*50	Parameters
Trainable weight	1,176,302	979,694	$\|\|\theta\|\|$
Project construction company simulation size	17	17	$proj$
Portfolio construction company simulation scaling	3	5	$port$
Number of convolution layers	2	2	n_k
Number of kernels	16	16	n_c
Convolution kernel width	5	5	s_k
Number of fully connected layers	3	3	f_k
Number of neurons	256	256	h_k

CRA: In this process, project sizes with different start and finish times and resource types are randomly sampled and layered, creating partially connected target projects. As an evaluation metric, the allocation of resources is traditionally used for CRA. This metric does not provide meaningful comparisons unless full coverage can be achieved. In this work, time delay which is constraining CRA is explored since full coverage is rarely feasible. Therefore, coverage ratio (CR) and coverage ratio and resources (CRAR) are defined as the evaluation metrics, which are both equally weighted if the ARA achieved success in resource allocation and zero otherwise. The CR is defined as the ratio of the number of target projects covered by the ARA to the total number of target projects in the construction company simulation. It measures the ARA's ability to successfully allocate resources and achieve coverage. The CRAR metric combines the CR with the consideration of time constraints. It assigns equal weight to both coverage and resource allocation success. The CRAR is calculated as follows:

$$CRAR = CR * ResourceAllocationSuccess \quad (23)$$

The CRAR metric is beneficial due to combining both goals, namely achieving high coverage and considering time constraints that impact target resources.

By utilizing randomly generated target projects as a baseline, the methodology enables meaningful comparisons of the ARA's performance across different scenarios. This approach allows for the evaluation and comparison of resource allocation strategies in dynamic construction environments, highlighting the effectiveness of the proposed methodology. As a result, the simulation provides an accurate assessment of the ARA's ability to achieve coverage while efficiently allocating resources within the given time constraints.

4.5. Data Harvesting

DH involves the ARA determining communication levels with devices by construction company simulation constant individual projects. The data rate is calculated using the same communications' channel parameters such as [21] for distance, random shadow fading, and LoS conditions. Similar to CRA, the size of the resources is not a relevant metric. Because of randomly changing connections, data amounts, and minimum time delays of IoT devices, all data cannot be collected in all scenarios. As a result, CR is used as the evaluation metric to describe the ratio of the obtained and accessible information, as soon as the information is gathered across all devices. Based on the CRA, the full information set and resource allocation performance are shown in one normalized metric by the CRAR in this context.

Overall, the motivation for DH lies in the need for a continuous and uninterrupted stream of information to facilitate proactive decision-making in dynamic construction environments. The continuous stream of data allows the RL model to adapt and learn optimal resource allocation strategies, addressing the dynamic nature of construction sites. Also, this paper introduces communication time slots to align the operation of communication systems with the decision time slots of the ARA system. This integration ensures that the collected data is efficiently utilized within the RL framework, emphasizing the synergy between data harvesting and RL-based decision-making. Equations such as Eq. 4 and Eq. 5 are instrumental in modeling information rate and signal-to-noise ratio, essential for effective data collection and transmission. The integration of these equations into our framework showcases the interplay between IoT data and RL processes in shaping resource allocation strategies. Through simulation and iterative RL processes, our ARA system learns and adapts based on real-time data. This iterative learning process is powered by the continuous data stream, demonstrating the indispensable role of data harvesting in creating the autonomous resource allocation.

5. Performance Evaluation

Approximately 20-50% of the available projects were covered by CRA agents trained on target projects containing 3-8 resources. A 'Site32' decision range of 50-150 steps was used, while a 'Site50' decision range of 150-250 steps was used. A DH scenario involves placing 3-10 devices in the project at random locations. Each device contains 5.0-20.0 data units. A detailed analysis of four scenarios is presented. According to Fig. 4 (a and b), the agents in the CRA scenarios are able to find most of the resources they need to complete the target project. There is some coverage for projects which do not require much resources. The coverage of small projects is incomplete since detours are avoided. While most of the target projects are efficiently

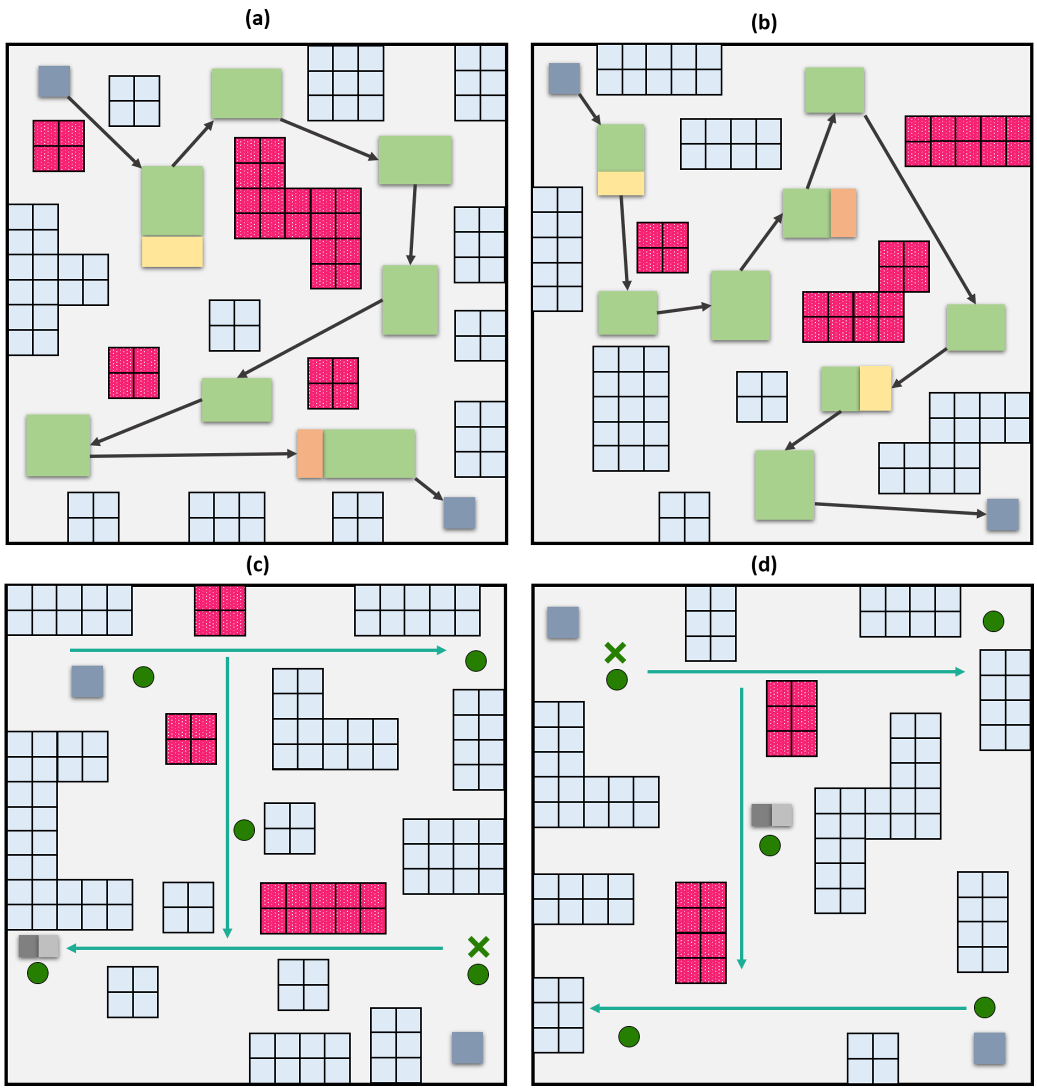

Figure 4: Examples of simulating CRA and DH scenarios using the Monte Carlo approach for the ARA system. The subfigure (a) and (b) illustrate CRA and DH of "Site32" scenarios, respectively. Likewise, subfigure (c) and (d) illustrate CRA and DH of "Site50" scenarios, respectively. The subfigure (c and d) shows excellent performance from the agents in the DH scenarios. This scenario results in a collection ratio of 99.1% as small bits of information are left at the green and gray devices by the agent.

covered, there are a few exceptions. Fig. 4 (c and d) shows excellent performance from the agents in the DH scenarios. This scenario results in a collection ratio of 99.1% as small bits of information are left at the green and gray devices by the agent. It uses only 92 out of 150 possible decisions to find a concise resource. 'Site50' demonstrates that the agent collected all the data and returned them by some decision steps left over. Two million steps were completed by all four agents. According to the results of a Monte Carlo simulation using 1000 Monte Carlo scenarios (Table 4), the allocation of resources performed by all four agents is good, but that of the Site50 agent is marginally better.

A representation of the ARA algorithm and scheduling optimized for the projects of a construction contractor is provided in Fig. 5. The Gantt chart is depicted for a construction portfolio containing several tasks of a number of ongoing projects. In fact, the company has six projects in its portfolio, including one that has been stopped completely. These projects are different in size, tasks, and required resources at the same time. Also, a number of tasks, including civil, electrical, and mechanical, are running in parallel while two have been stopped. In addition, three types of resources are available depending on the task type, namely human resources, material, and equipment. Based on the duration and immediate predecessors and successors, the determined tasks are scheduled, and Early Start (ES), Late Finish (LF), and float are defined. The amounts of available resources are reported daily through DH, collected from IoT across all projects. Then, the resources are allocated to active tasks considering their reported requests for different resources. This process is performed in a way that the projects would be completed in the shortest possible time without or with

Project	Task	Resource	R-Rqst	Schedule ES	Schedule LF	Schedule D	Schedule F	Status	Timeline (day) 1	2	3	4	5	6	7	8	9	10	11
1	Mechanical-Task 1	Human resource	4	1	8	4	4	Active									✗		
		Material	40														✗		
	Electrical-Task 3	Human resource	2	2	5	3	1	Active					✓						
		Material	10										✓						
2	Electrical-Task 5	-	-	2	8	5	2	Stopped											
	Civil-Task 7	Human resource	9	1	6	4	2	Active					✓	✓					
		Material	80										✓	✓					
		Equipment	3										✓	✓					
3	Mechanical-Task 3	Human resource	3	1	7	6	1	Active											
		Material	45																
	Electrical-Task 4	Human resource	3	2	6	3	2	Active						✓					
		Material	20											✓					
	Civil-Task 10	Human resource	6	2	8	4	3	Active											
		Material	90																
		Equipment	3																
4	Mechanical-Task 2	Human resource	5	2	6	4	1	Active							✗	✗			
		Material	55												✗	✗			
	Civil-Task 1	-	-	1	5	3	1	Stopped											
5	Electrical-Task 5	Human resource	4	3	8	4	2	Active											
		Material	35																
	Civil-Task 10	Human resource	7	1	7	6	1	Active								✗	✗	✗	✗
		Material	80													✗	✗	✗	✗
		Equipment	2													✗	✗	✗	✗
6	Civil-Task 3	-	-	2	6	3	2	Stopped											

Legend		Daily Available Resources:	1	2	3	4	5	6	7	8	9	10	11
R-Rqst	Resource Request	Human resource- Mechanical	10	10	8	9	7	5	10	12	11	12	9
ES	Early Start	Material- Mechanical	100	100	90	100	70	50	90	70	70	70	50
LF	Late Finish	Human resource- Electrical	0	5	6	5	3	4	5	6	7	9	6
D	Duration	Material- Electrical	0	30	60	60	60	70	70	60	50	40	30
F	Float	Human resource - Civil	18	17	17	19	16	20	19	19	18	19	8
✓	Earlier Finished	Material - Civil	250	250	180	180	170	150	150	140	100	100	90
✗	Delayed	Equipment	5	5	5	3	4	4	4	4	3	3	3

Figure 5: Autonomous resource allocation and optimized scheduling for a sample scenario in projects of a construction company.

Table 3
Legends for DH and CRA simulation plots of Fig. 4.

Mode	Description	Symbol
DDQN Input	Start and finishing point	
	Stopped projects	
	Projects' loss wireless links	
	IoT device	
	CRA: Remaining target project	
Visualization	DH: Summation of resource	
	DH: Updating while communicating with green device	→
	DH: Deciding while communicating with green device	✗
	CRA: Not covered and covered resource	
	Actions without communication	→

Table 4
Averaging of random scenario Monte Carlo simulations over 1000 iterations was used to determine performance metrics.

Metric	Site32: CRA	Site32: DH	Site50: CRA	Site50: DH
Due time	98.6%	98.4%	98.2%	99.4%
CR	71.8%	85.6%	83.5%	77.5%
CRAR	72.3%	83.5%	81.1%	77.2%

minimum delays and penalties. Furthermore, regarding the optimum resource utilization is one of the objectives of this model. That is to say, the selected project (s) for resource allocation every day should be the one(s) that use(s) as many available resources as possible, which helps with resource-leveling. Overall, Fig. 5 shows only an example of what the ARA model can do. Actually, this model can also be employed in larger or smaller companies. Moreover, the model can be utilized in every stage of the project, both from the beginning and the middle of the project as can be seen in Fig. 5.

In the Gantt chart as shown in Fig. 5, the best duration for the portfolio was determined by the proposed RL-based model, as expected. First, there were many violations of resource constraints in this portfolio, so intervention was required to remove them. However, doing these interventions manually would not be desirable, since it is very time-consuming especially when it is required immediately by the managers in meetings. In this case, manually solving a

Table 5
Flatten layer size for a comparison between two different strategies of simulation map processing, $port$ and $proj$.

Portfolio construction company simulation scaling $port$	Project construction company simulation scaling $proj$			
	9	17	25	33
2	8,581	9,751	13,189	18,455
3	2,751	4,003	7,339	12,704
5	274	1,543	4,882	10,254
7	33	1,323	4,631	10,015

Table 6
A comparison between two different strategies of simulation map processing in terms of the training time speedup.

Portfolio construction company	Project construction company simulation scaling $proj$							
	9		17		25		33	
simulation scaling $port$	CRA	DH	CRA	DH	CRA	DH	CRA	DH
2	2.7	2.3	2.3	2.0	1.8	1.5	1.2	1.1
5	3.4	2.9	3.1	2.4	2.0	1.8	1.6	1.4
7	4.3	3.5	3.4	3.1	2.5	2.3	1.8	1.6

small problem could take a lot of time. As such, it cannot be repeated several times to analyze different situations during a meeting. However, it is possible to produce similar results within seconds employing automated methods such as GA and RL. Indeed, both of them are capable of automatically generating resource allocation avoiding violating constraints. In forthcoming meetings, such algorithms can be used to instantly compute the effects of made decisions, in a short time. The GA takes a minute for computation, while the RL algorithm only takes a second. The problem becomes complicated when GA should be adapted to more complex portfolios. Actually, GA is expected to perform slower as the number of projects and resources increases and resources become more constrained. Nevertheless, it was found out that even when CPU time was taken into account, the RL algorithm performed significantly better than the GA, for generating solutions, in terms of computational time. In addition, the RL algorithm maintains its quality in producing optimal solutions regardless of scale or complexity, whereas the effectiveness of the genetic algorithm's result reduces. As a result, an RL-based approach always outperforms a GA-based approach with regard to processing time. Based on the achieved results, the proposed RL approach in this paper demonstrates its capability to support upcoming planning by allocating resources for different scenarios in real time. Therefore, it enables project managers to interactively assess the impact of different strategies and constraints on project duration and then make the optimal decision. As part of this method, the IoT and automated DH technologies are incorporated to assist project managers to plan ahead for error-free resource management.

5.1. Portfolio-Project Parameter Evaluation

Our study tested multiple situations by single agent with different variables on the CRA and DH problems in order to determine whether the new hyper-parameters have a significant impact on the performance. To test each possible combination, three agents for four values of $proj$ as well as four values of port were trained. Furthermore, three agents were trained without project and portfolio construction company simulation processing which would correspond to $proj = 0$ and $port = 1$. Based on 200 Monte Carlo scenarios, 51 agents were created for the CRA and DH problems. There was a difference between the previous assessment and this one because 150-300 decision steps were used. Following are the parameters selected according to (19) and the flatten layer size, as can be seen in Table 5. The CRAR values for the CRA and DH problems are shown in Fig. 4 (a and b), respectively, for each agent's flatten layer size. The training process is significantly faster than that for agents without portfolio and project simulation map processing. Parameters are more important in the DH problem than in the CRA problem, as can be seen in Table 6. Flattening layers are generally more effective up to a point when they have a larger thickness. CRAR can be zero for some runs for both problems because of a large flatten layer. In this case, the agent does not properly allocate resources due to a lack of learning. As DH agents don't use the portfolio-project simulation map approach, they have a very low CRAR score since they don't learn how to allocate consistently. Table 2 shows that the agents with $proj = 17$ and $port = 5$ or $port = 3$ present the best performance in both scenarios despite their small flatten layer sizes of only 33 neurons. Apart from these two parameter combinations, the agents with $port = 7$ and $proj = 9$ also perform well with these parameters.

A comparison of the time for resource allocation using various strategies is presented in Table 6, which shows that CRA and DH will be able to train with the proposed method of portfolio simulation processing faster than when no portfolio simulation processing is used. As a result, in the proposed method, there is no limitation in scheduling and allocating resources according to the number of projects. Although increasing the number of projects makes GA impractical due to the computation time, it will not change the computation time for resource allocation and project scheduling in the proposed method. Also, it can lead to non-optimal scheduling and allocation by the GA.

5.2. DRL Evaluation

We compare the performance of three RL algorithms, namely our approach without individual project information (A), our approach with individual project information (B), and our approach with individual project information and simulation processing (C), in the context of resource allocation in construction companies as shown in Fig. 6. The performance is evaluated based on Fig. 6 in terms of reward achieved at the different iterations of the training, the time required for convergence, and overall performance. The results provide insights into the effectiveness and weaknesses of these algorithms in improving resource allocation simulations in the construction industry.

Our approach without individual project information (A) demonstrates consistent improvement in reward over the training process, with reward values decreasing from -75 at the start to -15 at both the half and end training process. This indicates that A's resource allocation strategy effectively adapts and optimizes over time. However, the time required for convergence remains relatively high, decreasing from 90 units at the start to 40 units at the end. This suggests that A may have a slower convergence rate compared to other methods. Overall, A's performance can be considered slow in terms of convergence.

Our approach with individual project information (B) exhibits a similar pattern of reward improvement, though the magnitude of rewards is lower compared to A. The reward values decrease from -38 at the start to -12 at the half and -10 at the end. This suggests that B's resource allocation strategy is effective, but may not achieve the same level of optimization as A. In terms of convergence time, B shows a moderate performance, with a reduction in time from 90 units at the start to 38 units at the end. The moderate convergence assessment indicates that B's strategy may require additional iterations to reach an optimal solution.

Our approach with individual project information and simulation processing (C) stands out with significantly better reward values compared to both A and B. The reward decreases from -18 at the start to -7 at the half and further to -2 at the end. This indicates that C's resource allocation strategy outperforms the other methods in terms of optimizing project outcomes. However, the time required for convergence is relatively high, decreasing from 90 units at the start to 35 units at the end. This suggests that C may take faster to converge compared to other methods. Nevertheless, C achieves a high level of convergence, indicating its strong potential for effective resource allocation.

A demonstrates consistent improvement in reward throughout the project but has a relatively slower convergence rate. B achieves moderate rewards and convergence, suggesting room for further optimization. C outperforms both A and B in terms of reward but has a faster convergence time. These findings contribute to the understanding of RL methods for resource allocation in construction and provide valuable insights for decision-making in construction project management. Future research should focus on exploring the underlying mechanisms of C and further improving the performance of RL algorithms for resource allocation in construction.

Table 7 provides a comprehensive overview of the different simulated construction scenarios used to evaluate the ARA system in 'Site50' simulation. The scenarios are intentionally diverse to test performance across varied conditions. The number of projects ranges from just 5 in the simplest scenario to 40 in the most complex, with correspondingly increasing numbers of tasks per project from 5 to 80. This represents construction environments from smaller scale projects with few activities to large multifaceted programs.

The number of resource types also scales from 10 to 20 across the scenarios. Along with the growth in constraints of all types - precedence, discrete resource, disjunctive, logical, and parallel - this reflects the increasing complexity of constraints and dependencies that have to be navigated for successful resource allocation. At the low end, Scenario 1 has 30 precedence and 20 discrete resource constraints among its 5 projects and 10 resource types. At the high end, Scenario 8 has a robust 209 precedence constraints and 21 parallel constraints with its 40 projects, 80 tasks per project, and 20 and 40 resource types.

Table 8 provides insightful performance comparisons between the GA and DDQN approaches across the scenario testing. On project duration, DDQN outperforms GA by 3-583 days across the board, with its optimized allocation able to complete projects much faster than the GA benchmark. The advantages are even more pronounced on processing time, with DDQN finishing in seconds to minutes while GA takes minutes to over an hour. This demonstrates the computational efficiency gains of DDQN for resource optimization. Further, DDQN achieves CR 79-81% versus 67-75% for GA and CRAR of 80-82% versus 67-78% for GA. This quantifies the substantially higher coverage realized by DDQN for target project completion and resource allocation within time constraints.

In Figs. 7 and 8, we present the number of resources used per day for DDQN and GA across two representative simulation scenarios based on Table 7. Each scenario reflects a different level of complexity in terms of project size, task distribution, and resource constraints. As depicted in the figures, DDQN consistently achieves more efficient resource allocation compared to GA in both scenarios. Specifically, DDQN demonstrates a smoother and more balanced utilization of resources over time, leading to optimized project durations [80]. In contrast, GA exhibits more fluctuations and inefficiencies in resource usage, resulting in longer project

Table 7
Different simulation scenarios are described in detail.

Simulation Scenario	Number of Project	Number of Tasks per Projects	Number of Resource type	Number of Constraints of each type in the simulation scenario				
				Precedence	Discrete resource	Disjunctive	Logical	Parallel
1	5	5	10	30	20	4	10	10
2	5	10	20	58	40	6	10	12
3	10	10	10	82	20	9	10	5
4	10	20	20	110	40	4	10	9
5	20	20	10	69	20	5	10	14
6	20	30	20	123	40	5	10	14
7	40	40	10	167	20	8	10	18
8	40	80	20	209	40	9	10	21

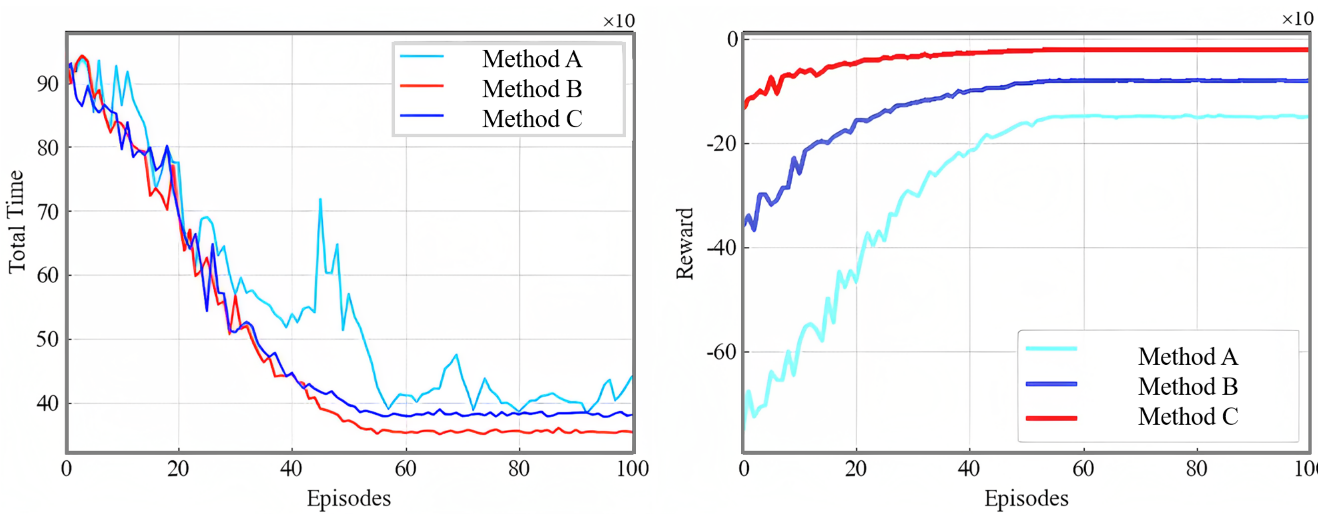

Figure 6: The effect of using simulation maps processing and project information on resource allocation and optimized scheduling. Method A is our approach without individual project information. Method B is our approach with individual project information, just including portfolio information. Method C is our proposed method with simulation information processing

Table 8
Comparison of performance of DDQN and GA in different simulated scenarios.

Simulation Scenario	Project duration (Day)		Processing time		CR(%)		CRAR(%)	
	GA	DDQN	GA	DDQN	GA	DDQN	GA	DDQN
1	30	27	53 s	3 s	75.42	80.54	77.23	80.39
2	77	69	1 m59 s	7 s	75.67	79.23	78.92	80.04
3	230	187	8 m46 s	19 s	73.82	79.82	74.11	80.46
4	405	236	14 m39 s	35 s	72.12	80.92	72.34	81.54
5	512	218	24 m48 s	54 s	71.02	80.67	70.71	82.16
6	696	305	30 m8ss	102 s	70.02	81.73	70.13	81.82
7	822	328	52 m29 s	151 s	68.89	81.32	68.51	82.43
8	957	374	1 h33 m38 s	205 s	67.82	80.33	67.11	80.58

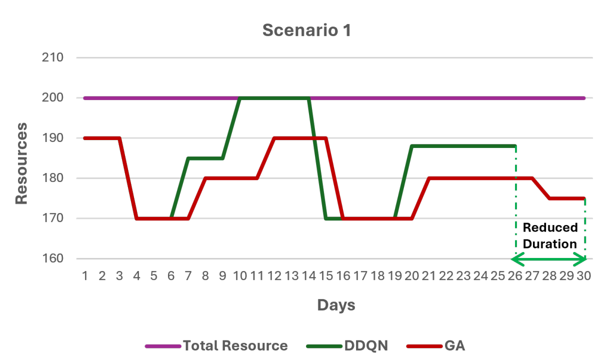

Figure 7: Comparison of resource allocation between DDQN and GA based on scenario 1 on Table 7 and the effect of resource allocation on project duration.

durations [80]. These findings align with the performance metrics presented in Table 8, where DDQN consistently outperforms GA in terms of project duration, CR, and CRAR. By visually demonstrating the superiority of DDQN in resource allocation efficiency, Figs. 7 and 8 provide further insight into why DDQN leads to shorter project durations compared to GA.

Together, the scenario testing highlights that DDQN's reinforcement learning approach delivers faster, more optimal, and effective resource allocation across diverse and complex simulated construction environments compared to traditional GA optimization. The results powerfully validate it as an exciting new technique for improving construction planning.

5.2.1. *Reward Function Evaluation*

Table 9 provides a comprehensive insight into the influence of different reward components on the performance of the ARA system in two construction company simulations (Site32 and Site50). The positive reward component (r_c) consistently demonstrates its significance in enhancing the Due Time metric, showcasing its efficacy in promoting the timely completion of projects. Moreover, this positive

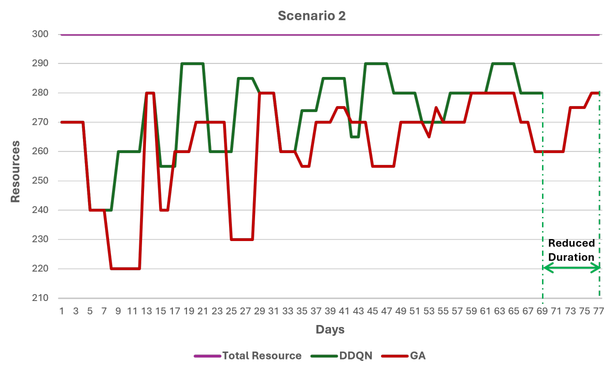

Figure 8: Comparison of resource allocation between DDQN and GA based on scenario 2 on Table 7 and the effect of resource allocation on project duration.

Table 9
Effect of Reward Function Components on Performance Metrics.

Reward Component	Metric	Site32: CRA	Site32: DH	Site50: CRA	Site50: DH
r_c (Positive)	Due Time	95.6%	96.4%	95.2%	96.4%
	CR	70.8%	83.3%	81.7%	75.7%
	CRAR	71.3%	81.5%	79.1%	76.6%
r_{sc} (Negative)	Due Time	94.5%	95.8%	94.9%	95.2%
	CR	70.2%	83.9%	82.1%	76.8%
	CRAR	70.5%	82.7%	80.5%	76.1%
r_{dec} (Negative)	Due Time	96.8%	95.2%	95.9%	96.5%
	CR	71.5%	84.2%	80.4%	76.2%
	CRAR	71.1%	82.1%	81.2%	75.8%
r_{delay} (Negative)	Due Time	99.2%	99.1%	99.2%	99.7%
	CR	68.1%	74.0%	77.2%	71.3%
	CRAR	69.0%	76.5%	77.4%	73.0%
$r_c + r_{\text{delay}}$	Due Time	98.4%	98.1%	99.0%	99.5%
	CR	70.5%	84.8%	82.9%	77.3%
	CRAR	71.0%	83.3%	81.5%	76.4%
$r_c + r_{sc}$	Due Time	97.9%	98.0%	98.1%	98.4%
	CR	71.0%	84.2%	82.3%	77.1%
	CRAR	71.8%	82.8%	80.8%	76.3%
$r_c + r_{dec}$	Due Time	98.1%	98.3%	98.4%	98.7%
	CR	71.3%	84.5%	82.7%	77.2%
	CRAR	71.7%	83.1%	81.0%	76.5%
$r_c + r_{dec} + r_{sc} + r_{\text{delay}}$	Due Time	98.6%	98.4%	98.2%	99.4%
	CR	71.8%	85.6%	83.5%	77.5%
	CRAR	72.3%	83.5%	81.1%	77.2%

impact extends to the CR and CRAR, signifying its role in improving resource allocation success and overall coverage.

Conversely, the negative safety controller penalty (r_{sc}) introduces a subtle dynamic. While it marginally reduces the Due Time metric, it concurrently leads to improvements in CR and CRAR. This suggests that penalizing the ARA for colliding with projects or resource constraints serves to enhance safety and resource allocation success, thus highlighting the delicate trade-offs involved in balancing safety considerations with efficiency.

Similarly, the negative decision penalty (r_{dec}) exhibits a minor reduction in the Due Time metric but contributes positively to CR and CRAR. This outcome suggests that penalizing incomplete tasks encourages the ARA to prioritize and make decisions efficiently, aligning with the overall goal of optimal resource allocation.

The most significant impact is observed with the negative delay penalty (r_{delay}), which significantly improves the Due Time metric. This penalty effectively incentivizes the ARA to manage time more judiciously, resulting in timely project completion. Moreover, CR and CRAR benefit from this penalty, underscoring the crucial role of time management in resource allocation success.

The combinations of these individual reward components provide further insights into the delicate balance required for an effective reward function. The combination of $r_c + r_{\text{delay}}$ maintains high Due Time metrics while improving CR and CRAR, suggesting a balanced approach between meeting deadlines and efficient resource allocation. Combining $r_c + r_{sc}$ demonstrates positive impacts across all metrics, emphasizing the importance of considering both positive rewards and safety constraints for optimal performance. Meanwhile, the combination of $r_c + r_{dec}$ strikes a balance, leading to improvements in CR and CRAR without compromising Due Time significantly. The comprehensive combination of all reward components in $r_c + r_{dec} + r_{sc} + r_{\text{delay}}$ yields high Due Time metrics and well-balanced improvements in CR and CRAR, underscoring the efficacy of considering all aspects in the reward function for a holistic and effective performance. Overall, the proposed reward function, as encapsulated in the last row of the table, showcases not only the novelty but also the efficacy of integrating diverse elements to achieve a comprehensive and balanced performance across various construction scenarios. This approach considers safety, resource allocation success, decision-making efficiency, and timely project completion, making it a robust and adaptable framework for Autonomous Resource Allocation in dynamic construction environments.

6. Discussions

Similar to any other research project, this study has some limitations. However, our main motivation was to address the limitations of existing approaches in resource management. DRL's design, our proposed model, allows us to model the system dynamics smoothly and capture the complex interactions among features of resource allocation.

One limitation of the proposed ARA system in this paper is that it does not consider the detailed aspects of simulation, such as transportation logistics and resource preparation. While the system incorporates CRA and DH methods to optimize resource allocation, it does not fully capture the intricacies involved in transportation planning or the preparation of resources for construction projects.

Transportation logistics play a significant role in efficient resource management. Factors such as the availability of transportation modes, route optimization, and scheduling are crucial for the timely delivery of resources to the project sites. Ignoring these transportation details may result in suboptimal allocation and delays in resource deployment, ultimately affecting the overall efficiency of the construction company.

Additionally, the preparation of resources, such as prefabrication or assembly of components off-site, can significantly impact project timelines and resource utilization. Failing to consider these aspects in the simulation may limit the system's ability to optimize resource allocation comprehensively.

The simplification of data collection in our simulation is a deliberate choice aimed at maintaining the simplification of the simulation. As indicated in Table 8, our model demonstrates a commendable level of generalizability even as the number of projects and resources increases. This generalizability implies that the model's effectiveness is not compromised by the simplified data collection approach. Importantly, this robust performance holds true even as the system navigates diverse scenarios, showcasing its ability to adapt to complex resource allocation dynamics. The inclusion of additional constraints such as transportation logistics and resource preparation would undoubtedly enhance the realism of the simulation, we view these aspects as avenues for future research. The current study focuses on the macro-level optimization of resource allocation, and the deliberate simplification allows us to emphasize the core principles of our innovative ARA system. The decision to add more constraints, including those related to preparation and transportation, can be considered in subsequent research to further refine and expand the integrated resource allocation system.

Although our suggested ARA system presents an innovative approach for macro-level resource allocation optimization, it also recognizes the necessity of a smooth integration with project-specific knowledge. In order to achieve this integration, the decision-making and communication pathways between the micro and macro levels of project management must be carefully considered. Analyzing the intricacies of this collaboration, we recognize the importance of empowering project managers with tools to input project-specific constraints and preferences into the ARA system. Through the provision of an interface, project managers may offer their sophisticated comprehension of issues encountered in the field, enabling the system to modify its optimization tactics to more closely conform to the specifics of each project. The ARA system's practical usefulness is reinforced by this analytical cooperation, which not only improves resource allocation accuracy but also cultivates a sense of ownership and participation among project managers. Moreover, this collaborative framework's creation of a feedback loop is essential. The ARA system facilitates regular contacts between portfolio management and project managers, which allow for ongoing learning and adaptation. The system may be improved over time by feeding back dynamic changes, unanticipated obstacles and issues unique to the project. This analytical feedback loop ensures that the ARA system remains responsive to the ever-changing demands of construction projects, ultimately enhancing its efficacy and relevance in real-world scenarios. Overall, the bridge between project managers and portfolio management in the context of the ARA system extends beyond a mere exchange of information; it is a symbiotic relationship that leverages the strengths of both micro and macro perspectives. By empowering project managers with decision-making input and establishing a feedback loop, our methodology not only optimizes resource allocation but also creates a collaborative ecosystem that thrives on adaptability and continuous improvement.

DRL models, including the one proposed in our study, strike a delicate balance between exploration and exploitation during the learning process. This balance allows the model to systematically explore a wide range of potential resource allocation strategies while exploiting learned knowledge to refine the decision-making process. By maintaining this equilibrium, our ARA system significantly reduces the risk of being trapped in local optima and, instead, converges toward more globally optimal solutions. Also, our paper presents a thorough set of experimental results that reinforce the efficacy of our approach in achieving global optima. We demonstrate the ARA system's consistent performance across diverse scenarios, substantiating its ability to navigate complex resource allocation dynamics and providing evidence of its resilience against local optima. These experimental validations serve to emphasize the robustness and practical applicability of our proposed methodology.

There is no explicit or rigorous mathematical proof for DRL models, due to their black box feature extraction nature. However, our paper presents a comprehensive set of experimental results that validate the performance of our proposed method. Objective Key Performance Indicators (KPIs) such as CRA and DH were employed to assess the effectiveness of our model across various scenarios. Extensive experiments were conducted to demonstrate the efficiency of our approach in handling resource management portfolios and accurately allocating resource. Additionally, we conducted a detailed analysis to understand the contribution of different sizes of projects to the overall performance. These analyses, complemented by visualizations, provide compelling evidence supporting the effectiveness of DRL in resource management tasks.

7. Conclusions

As the construction industry is highly competitive, optimal resource management is one of the most important roles of the project manager. Computer application software offers the advantage of accurate resource management which distinguishes them from manual methods. The main objective of this paper is to present a general solution for Autonomous Resource Allocation (ARA) that can be applied to two distinct objectives, coverage resource allocation (CRA) and data harvesting (DH). The reward function was designed to combine specific objectives with decision constraints so that DDQNs could efficiently assign resources in both scenarios. In this paper, a novel portfolio-project simulation map processing scheme was presented that determines how simulation map parameters affect the learning performance of the DRL agent, allowing large simulation maps to be directly fed into convolutional layers.

The proposed model is beneficial for numerous projects of different sizes and with different amounts of resources. In spite of previous methods such as the GA, this model eliminates the computational weaknesses in terms of time and complexity. In fact, a few seconds are required to perform calculations of optimal resource allocation, resource leveling, and scheduling of projects. Furthermore, the proposed model is designed in such a way that it can schedule the rest of the tasks from any stage of the project. Due to the very short time required for doing calculations and providing results by this model, it can be used in every phase of the project with the improvements of the model in the future. For instance, using this model in decision-making meetings enables senior managers to make the optimal decision.

The next step is to examine the remaining hindrances to the application of our method to even larger construction company simulation, namely micro-alternation of decisions through macro-actions or options. In the future, it will also be possible to perform experiments with realistic ARA simulators using the combination of the presented high-level resource allocation approach and the low-level time delay controller. The performance of resource allocation will be further examined by investigating irregular projects and non-convex obstacles. Also, future research explore integrating transportation planning algorithms and resource preparation strategies within the ARA system to overcome these limitations. By incorporating these details into the simulation and decision-making processes, the system would have a more holistic view of resource management, enabling it to make more accurate and efficient allocation decisions.

8. Disclosures

The authors declare no conflict of interest.

9. Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 2222881. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] A. H. Lamka[1], S. M. Masu, G. Wanyona, Towards an appropriate construction industry resource levelling model for kenya, International Journal of Engineering Research and Technology 7 (9) (9 2018). doi:10.17577/ijertv7is090049.

[2] M. Eizeldin, M. H. Hegazy, A. Alhady, S. Ashraf, Analyzing resource allocation and levelling in construction projects, American Journal of Engineering Research 11 (4) (2022) pp. 108–117, [Accessed Feb, 10, 2024].
URL https://www.ajer.org/papers/Vol-11-issue-4/K1104108117.pdf

[3] S. Biruk, A computer-based renewable resource management system for a construction company, Open Engineering 8 (1) (2018) pp. 440–446. doi:10.1515/eng-2018-0062.

[4] N. Dayoub, M. Fakhratov, The mutual influence approach during the resource allocation process in construction projects, in: IOP Conference Series: Materials Science and Engineering, Vol. 1030, IOP Publishing, 2021, p. 012104. doi:10.1088/1757-899x/1030/1/012104.

[5] H. Li, G. Chan, M. Skitmore, T. Huang, A 4d automatic simulation tool for construction resource planning: a case study, Engineering, Construction and Architectural Management 22 (5) (2015) pp. 536–550. doi:10.1108/ecam-07-2014-0093.

[6] D. M. Hall, I. Čustović, R. Sriram, Q. Chen, Teaching generative construction scheduling: Proposed curriculum design and analysis of student learning for the tri-constraint method, Advanced Engineering Informatics 51 (2022) p. 101455. doi:10.1016/j.aei.2021.101455.

[7] D. K. Chua, L. J. Shen, Key constraints analysis with integrated production scheduler, Journal of Construction Engineering and Management 131 (7) (2005) pp. 753–764. doi:10.1061/(ASCE)0733-9364(2005)131:7(753).

[8] K. Kim, J. M. de la Garza, Phantom float, Journal of Construction Engineering and Management 129 (5) (2003) pp. 507–517. doi:10.1061/(ASCE)0733-9364(2003)129:5(507).

[9] P. Dallasega, E. Marengo, A. Revolti, Strengths and shortcomings of methodologies for production planning and control of construction projects: a systematic literature review and future perspectives, Production Planning & Control 32 (4) (2021) pp. 257–282. doi:10.1080/09537287.2020.1725170.

[10] B. Huber, P. Reiser, The marriage of cpm and lean construction, in: 11th Annual Conference of the International Group for Lean Construction, 2003, [Accessed Feb, 10, 2024].
URL https://iglc.net/papers/Details/241

[11] S. Münker, P. R. Wildemann, A. Göppert, S. Brell-Cokcan, R. H. Schmitt, Online capability-based resource allocation for on-site construction operations utilizing digital twin models, Construction Robotics 5 (3) (2021) pp. 211–226. doi:10.1007/s41693-022-00065-4.

[12] M. A. Eirgash, Resource allocation and leveling in construction management projects with resource histogram, American Journal of Engineering and Technology Management 5 (6) (2020) pp. 91–95. doi:10.11648/j.ajetm.20200506.11.

[13] D. K. Chua, L. Shen, S. Bok, Constraint-based planning with integrated production scheduler over internet, Journal of Construction Engineering and Management 129 (3) (2003) pp. 293–301. doi:10.1061/(ASCE)0733-9364(2003)129:3(293).

[14] J. J. Shi, D. W. Halpin, Enterprise resource planning for construction business management, Journal of Construction Engineering and Management 129 (2) (2003) pp. 214–221. doi:10.1061/(ASCE)0733-9364(2003)129:2(21.

[15] P.-H. Chen, H. Weng, A two-phase ga model for resource-constrained project scheduling, Automation in Construction 18 (4) (2009) pp. 485–498. doi:10.1016/j.autcon.2008.11.003.

[16] E. Sriprasert, N. Dawood, Multi-constraint information management and visualisation for collaborative planning and control in construction, Journal of Information Technology in Construction 8 (2003) pp. 341–366, [Accessed Feb, 10, 2024].
URL https://www.itcon.org/paper/2003/25

[17] A. R. Carrel, P. L. Palmer, An evolutionary algorithm for near-optimal autonomous resource management, in: 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Vol. 603, 2005, p. 25, [Accessed Feb, 10, 2024].
URL https://www.researchgate.net/publication/228357374_An_evolutionary_algorithm_for_near-optimal_autonomous_resource_management

[18] H. Shen, L. Chen, Distributed autonomous virtual resource management in datacenters using finite-markov decision process, IEEE/ACM Transactions on Networking 25 (6) (2017) pp. 3836–3849. doi:10.1109/tnet.2017.2759276.

[19] S. Tang, B. He, H. Liu, B.-S. Lee, Chapter 7 - resource management in big data processing systems, Big Data Principles and Paradigm (2016) pp. 161–188doi:10.1016/B978-0-12-805394-2.00007-6.

[20] S. Bin, G. Sun, Optimal energy resources allocation method of wireless sensor networks for intelligent railway systems, Sensors 20 (2) (2020) p. 482. doi:10.3390/s20020482.

[21] H. Bayerlein, M. Theile, M. Caccamo, D. Gesbert, Uav path planning for wireless data harvesting: A deep reinforcement learning approach, in: 2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6. doi:10.1109/globecom42002.2020.9322234.

[22] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, A. Corradi, Efficient data harvesting in mobile sensor platforms, in: Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, IEEE, 2006, pp. 351–356. doi:10.1109/percomw.2006.47.

[23] M. Loosemore, A. Dainty, H. Lingard, Human resource management in construction projects: strategic and operational approaches, Routledge, 2003. doi:10.4324/9780203417881.

[24] M. P. Wylie, J. Holtzman, The non-line of sight problem in mobile location estimation, in: 5th International Conference on Universal Personal Communications, Vol. 2, IEEE, 1996, pp. 827–831. doi:10.1109/icupc.1996.562692.

[25] X. Deng, P. Guan, C. Hei, F. Li, J. Liu, N. Xiong, An intelligent resource allocation scheme in energy harvesting cognitive wireless sensor networks, IEEE Transactions on Network Science and Engineering 8 (2) (2021) pp. 1900–1912. doi:10.1109/tnse.2021.3076485.

[26] I. Ahmed, M. M. Butt, C. Psomas, A. Mohamed, I. Krikidis, M. Guizani, Survey on energy harvesting wireless communications: Challenges and opportunities for radio resource allocation, Computer Networks 88 (2015) pp. 234–248. doi:10.1016/j.comnet.2015.06.009.

[27] H. Al-Tous, I. Barhumi, Differential game for resource allocation in energy harvesting wireless sensor networks, IEEE Transactions on Green Communications and Networking 4 (4) (2020) pp. 1165–1173. doi:10.1109/tgcn.2020.3009268.

[28] S. S. Kalamkar, J. P. Jeyaraj, A. Banerjee, K. Rajawat, Resource allocation and fairness in wireless powered cooperative cognitive radio networks, IEEE Transactions on Communications 64 (8) (2016) pp. 3246–3261. doi:10.1109/tcomm.2016.2581162.

[29] X. Li, L. Da Xu, A review of internet of things—resource allocation, IEEE Internet of Things Journal 8 (11) (2020) pp. 8657–8666. doi:10.1109/jiot.2020.3035542.

[30] G. Tesauro, N. K. Jong, R. Das, M. N. Bennani, A hybrid reinforcement learning approach to autonomic resource allocation, in: 2006 IEEE International Conference on Autonomic Computing, IEEE, 2006, pp. 65–73. doi:10.1109/icac.2006.1662383.

[31] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, An introduction to deep reinforcement learning, Foundations and Trends in Machine Learning 11 (3-4) (2018) pp. 219–354. doi:10.1561/9781680835397.

[32] A. Angelova, L. Matthies, D. Helmick, P. Perona, Learning and prediction of slip from visual information, Journal of Field Robotics 24 (3) (2007) pp. 205–231. doi:10.1002/rob.20179.

[33] R. O. Afolabi, A. Dadlani, K. Kim, Multicast scheduling and resource allocation algorithms for ofdma-based systems: A survey, IEEE Communications Surveys & Tutorials 15 (1) (2012) pp. 240–254. doi:10.1109/surv.2012.013012.00074.

[34] M. Klinkowski, P. Lechowicz, K. Walkowiak, Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking, Optical Switching and Networking 27 (2018) pp. 58–78. doi:10.1016/j.osn.2017.08.003.

[35] M. Patriksson, A survey on the continuous nonlinear resource allocation problem, European Journal of Operational Research 185 (1) (2008) pp. 1–46. doi:10.1016/j.ejor.2006.12.006.

[36] Y. Xie, Z. Liu, S. Wang, Y. Wang, Service function chaining resource allocation: A survey, arXiv preprint (7 2016). doi:10.48550/arXiv.1608.00095.

[37] C. Carbno, Optimal resource allocation for projects, Project Management Journal 30 (2) (1999) pp. 22–31. doi:10.1177/875697289903000205.

[38] J. A. Petersen, V. Kumar, Perceived risk, product returns, and optimal resource allocation: Evidence from a field experiment, Journal of Marketing Research 52 (2) (2015) pp. 268–285. doi:10.1509/jmr.14.0174.

[39] F. Fiedrich, F. Gehbauer, U. Rickers, Optimized resource allocation for emergency response after earthquake disasters, Safety Science 35 (1-3) (2000) pp. 41–57. doi:10.1016/s0925-7535(00)00021-7.

[40] W. He, W. Li, W. Wang, Developing a resource allocation approach for resource-constrained construction operation under multi-objective operation, Sustainability 13 (13) (2021) p. 7318. doi:10.3390/su13137318.

[41] H. Zhang, H. Li, Simulation-based optimization for dynamic resource allocation, Automation in Construction 13 (3) (2004) pp. 409–420. doi:10.1016/j.autcon.2003.12.005.

[42] Y. Zeng, Q. Wu, R. Zhang, Accessing from the sky: A tutorial on uav communications for 5g and beyond, Proceedings of the IEEE 107 (12) (2019) pp. 2327–2375. doi:10.1109/jproc.2019.2952892.

[43] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, Association for the Advancement of Artificial Intelligence, 2016, pp. 2094–2100. doi:10.1609/aaai.v30i1.10295.

[44] F. B. Talbot, Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case, Management Science 28 (10) (1982) pp. 1197–1210. doi:10.1287/mnsc.28.10.1197.

[45] B. Gavish, H. Pirkul, Algorithms for the multi-resource generalized assignment problem, Management Science 37 (6) (1991) pp. 695–713. doi:10.1287/mnsc.37.6.695.

[46] J. P. Stinson, A branch and bound algorithm for a general class of resource-constrained scheduling problems, Ph. D. Dissertation, University of North Carolina at Chapel Hill (1976) pp. 252–259[Accessed Feb, 10, 2024].
URL https://www.proquest.com/docview/302815285?pq-origsite=gscholar&fromopenview=true

[47] W.-T. Chan, D. K. Chua, G. Kannan, Construction resource scheduling with genetic algorithms, Journal of Construction Engineering and Management 122 (2) (1996) pp. 125–132. doi:10.1061/(ASCE)0733-9364(1996)122:2(125).

[48] S.-S. Leu, C.-H. Yang, Ga-based multicriteria optimal model for construction scheduling, Journal of Construction Engineering and Management 125 (6) (1999) pp. 420–427. doi:10.1061/(ASCE)0733-9364(1999)125:6(420.

[49] F. F. Boctor, Some efficient multi-heuristic procedures for resource-constrained project scheduling, European Journal of Operational Research 49 (1) (1990) pp. 3–13. doi:10.1016/0377-2217(90)90116-s.

[50] E. M. Padilla, R. I. Carr, Resource strategies for dynamic project management, Journal of Construction Engineering and Management 117 (2) (1991) pp. 279–293. doi:10.1061/(ASCE)0733-9364(1991)117:2(279).

[51] C. E. Bell, J. Han, A new heuristic solution method in resource-constrained project scheduling, Naval Research Logistics 38 (3) (1991) pp. 315–331. doi:10.1002/1520-6750(199106)38:3<315::AID-NAV3220380304>3.0.CO;2-7.

[52] S. E. Sampson, E. N. Weiss, Local search techniques for the generalized resource constrained project scheduling problem, Naval Research Logistics 40 (5) (1993) pp. 665–675. doi:10.1002/1520-6750(199308)40:5<665::AID-NAV3220400509>3.0.CO;2-J.

[53] E. W. Davis, J. H. Patterson, A comparison of heuristic and optimum solutions in resource-constrained project scheduling, Management Science 21 (8) (1975) pp. 944–955. doi:10.1287/mnsc.21.8.944.

[54] J.-K. Lee, Y.-D. Kim, Search heuristics for resource constrained project scheduling, Journal of the Operational Research Society 47 (5) (1996) pp. 678–689. doi:10.1057/jors.1996.79.

[55] K. Bouleimen, H. Lecocq, A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version, European Journal of Operational Research 149 (2) (2003) pp. 268–281. doi:10.1016/s0377-2217(02)00761-0.

[56] E. Pinson, C. Prins, F. Rullier, Using tabu search for solving the resource-constrained project scheduling problem, in: International Workshop on Project Management and Scheduling, 1994, pp. 102–106, [Accessed Feb, 10, 2024].
URL https://hal.science/hal-02895382/

[57] T. Baar, P. Brucker, S. Knust, Tabu search algorithms and lower bounds for the resource-constrained project scheduling problem, in: Meta-heuristics, Springer, 1999, pp. 1–18. doi:10.1007/978-1-4615-5775-3_1.

[58] V. J. Leon, R. Balakrishnan, Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling, Operations-Research-Spektrum 17 (2) (1995) pp. 173–182. doi:10.1007/bf01719262.

[59] J. Alcaraz, C. Maroto, A robust genetic algorithm for resource allocation in project scheduling, Annals of Operations Research 102 (1) (2001) pp. 83–109. doi:10.1023/a:1010949931021.

[60] S. Hartmann, A competitive genetic algorithm for resource-constrained project scheduling, Naval Research Logistics 45 (7) (1998) pp. 733–750. doi:10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C.

[61] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, IEEE Transactions on Neural Networks 9 (5) (1998) p. 1054. doi:10.1109/tnn.1998.712192.

[62] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, Journal of Artificial Intelligence Research 4 (1996) pp. 237–285. doi:10.1613/jair.301.

[63] H. Zhang, H. Li, Simulation-based optimization for dynamic resource allocation, Automation in Construction 13 (3) (2004) pp. 409–420. doi:10.1016/j.autcon.2003.12.005.

[64] Y. Liu, S.-l. Zhao, X.-k. Du, S.-q. Li, Optimization of resource allocation in construction using genetic algorithms, in: 2005 International Conference on Machine Learning and Cybernetics, Vol. 6, IEEE, 2005, pp. 3428–3432. doi:10.1109/icmlc.2005.1527534.

[65] M. El-Gafy, Resource allocation for repetitive construction schedules: An ant colony optimization approach, in: Proceedings of the ASC 43rd Annual International Conference, Flagstaff, Arizona, [Accessed Feb, 10, 2024].
URL https://www.academia.edu/2755469/Resource_Allocation_for_Repetitive_Construction_Schedules_An_Ant_Colony_Optimization_Approach?sm=b

[66] J.-w. Huang, X.-x. Wang, R. Chen, Genetic algorithms for optimization of resource allocation in large scale construction project management, Journal of Computers 5 (12) (2010) pp. 1916–1924, [Accessed Feb, 10, 2024]. doi:10.4304/jcp.5.12.1916-1924.
URL http://www.jcomputers.us/index.php?m=content&c=index&a=show&catid=136&id=2331

[67] M. Rostami, M. Bagherpour, Optimization of multi period-multi location construction projects considering resource pool and batch ordering, International Journal of Optimization in Civil Engineering 9 (1) (2019) pp. 107–127, [Accessed Feb, 10, 2024].
URL https://ijoce.iust.ac.ir/article-1-378-en.html

[68] V. Asghari, A. J. Biglari, S.-C. Hsu, Multiagent reinforcement learning for project-level intervention planning under multiple uncertainties, Journal of Management in Engineering 39 (2) (2023) p. 04022075. doi:10.1061/JMENEA.MEENG-488.

[69] C. Jiang, X. Li, J.-R. Lin, M. Liu, Z. Ma, Adaptive control of resource flow to optimize construction work and cash flow via online deep reinforcement learning, Automation in Construction 150 (2023) p. 104817. doi:10.1016/j.autcon.2023.104817.

[70] O. Kammouh, M. Nogal, R. Binnekamp, A. Wolfert, Dynamic control for construction project scheduling on-the-run, Automation in Construction 141 (2022) p. 104450. doi:10.1016/j.autcon.2022.104450.

[71] 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) p. 104069. doi:10.1016/j.autcon.2021.104069.

[72] K. Lawal, H. N. Rafsanjani, Trends, benefits, risks, and challenges of iot implementation in residential and commercial buildings, Energy and Built Environment 3 (3) (2022) pp. 251–266. doi:10.1016/j.enbenv.2021.01.009.

[73] M. Jia, A. Komeily, Y. Wang, R. S. Srinivasan, Adopting internet of things for the development of smart buildings: A review of enabling technologies and applications, Automation in Construction 101 (2019) pp. 111–126. doi:10.1016/j.autcon.2019.01.023.

[74] J. Xie, L. R. G. Carrillo, L. Jin, An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems, IEEE Control Systems Letters 3 (1) (2018) pp. 67–72. doi:10.1109/lcsys.2018.2851661.

[75] L. P. Kaelbling, M. L. Littman, A. R. Cassandra, Planning and acting in partially observable stochastic domains, Artificial Intelligence 101 (1-2) (1998) pp. 99–134. doi:10.1016/s0004-3702(98)00023-x.

[76] R. S. Sutton, A. G. Barto, R. J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Systems Magazine 12 (2) (1992) pp. 19–22. doi:10.23919/acc.1991.4791776.

[77] F. Garcia, E. Rachelson, Markov decision processes, Markov Decision Processes in Artificial Intelligence (2013) pp. 1–38.doi:10.1002/9781118557426.ch1.

[78] S. Zhang, R. S. Sutton, A deeper look at experience replay, arXiv preprint (12 2017). doi:10.48550/arXiv.1712.01275.

[79] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning, Omnipress, 2010, pp. 807–814, [Accessed Feb, 10, 2024].
URL https://icml.cc/Conferences/2010/papers/432.pdf

[80] L. Liu, S. A. Burns, C.-W. Feng, Construction time-cost trade-off analysis using lp/ip hybrid method, Journal of Construction Engineering and Management 121 (4) (1995) pp. 446–454. doi:10.1061/(ASCE)0733-9364(1995)121:4(446).