SCHEDULING OPTIMIZATION OF ELECTRIC READY MIXED CONCRETE VEHICLES USING AN IMPROVED MODEL-BASED REINFORCEMENT LEARNING

Zhengyi Chen, Changhao Song, Xiao Zhang & Jack C.P. Cheng*

Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

ABSTRACT: *Decarbonizing the construction sector has become an imperative global agenda, with electric machinery playing a pivotal role in realizing this objective. This research concentrates on devising an operational scheduling optimization method for electric ready-mixed concrete vehicles (ERVs) – a groundbreaking, eco-friendly intervention for the construction sector. We commence by outlining a systematic problem definition for the ERV operational process, considering the distinctive characteristics of electric vehicles and ready-mixed concrete (RMC) delivery tasks. The entire process is then conceptualized as a Markov decision problem (MDP), which enables sequential decision-making. We subsequently develop an enhanced model-based reinforcement learning technique, named parallel-masked-decaying Monte Carlo Tree Search (PMD-MCTS), for efficient resolution of the MDP. The entire system is authenticated via a real-world case study, and the PMD-MCTS's performance is juxtaposed against existing benchmarks. The results demonstrate the appropriateness of the proposed MDP formulation for tackling RMC delivery tasks. The PMD-MCTS algorithm and one of its ablation algorithms (PM-MCTS) have demonstrated superior performance compared to other benchmarks in either cost reduction or delay minimization, with PMD-MCTS requiring 30% less computation time than PM-MCTS.*

KEYWORDS: Electric vehicle, Ready-mixed concrete delivery; Scheduling optimization; Model-based reinforcement learning; Monte Carlo Tree Search

1. INTRODUCTION

The escalating issue of carbon emissions, largely attributing to global warming, has necessitated decarbonization as a global imperative for sustainable development (Sinha & Chaturvedi, 2019). As a result, decarbonization has emerged as a global priority for sustainable development (Bogachkova, Guryanova, & Usacheva, 2022). Construction industry activities are a significant source of environmental pollution, responsible for approximately one-third of carbon emissions (Gan, Chan, Tse, Lo, & Cheng, 2017). Specifically, the ready-mixed concrete (RMC) production accounts for a large portion of global emissions (Olanrewaju, Edwards, & Chileshe, 2020). In addition, RMC is a still-growing market due to the rise of green building construction and the urbanization in developing countries (Hart, Nilsson, & Raphael, 1968). Palaniappan, Bashford, Li, Fafitis, and Stecker (2009) indicated that the transportation of RMC represents a major component of energy use and emissions. Therefore, optimizing the scheduling of RMC delivery is crucial for a greener construction industry. Due to the significant development of battery technology and automation, the electric drive technology has been regarded as a promising solution for improving the sustainability of the construction industry (T. Lin et al., 2020). Truck manufacturers have recently developed several electric RMC trucks that aim to implement emission-free transport in the construction industry (*Volvo Trucks delivers the first heavy-duty electric concrete mixer truck to CEMEX*, 2023). However, the academic focus on construction electric vehicles (CEVs), a cross-domain technology integrating the unique properties of electric vehicles (EVs) and the construction industry, has been limited. This research aims to bridge this gap by focusing on the scheduling optimization of electric ready-mixed concrete vehicles (ERV) to further the cause of greener construction.

In the EV domain, several battery-related factors have been extensively studied, such as battery status, charging rates and prices, and charging station locations (Turan, Pedarsani, & Alizadeh, 2020). However, most EV-related studies are not directly applicable to the construction industry due to its unique characteristics and requirements. Furthermore, existing CEV-related studies mainly focus on hardware improvements such as drivetrain (Tong, Jiang, Tong, Zhang, & Wu, 2023) and transmission system (Tan, Yang, Zhao, Hai, & Zhang, 2018), with little attention paid to the management-related topics. Studies related to RMC production and delivery have primarily focused on developing optimization formulations and optimization algorithms. For instance, P.-C. Lin, Wang, Huang, and Wang (2010) formulated the RMC delivery as a job shop problem, where each RMC delivery represents a job operation carried out by one of the trucks that correspond to the workstations. Z. Liu, Zhang, Yu, and Zhou (2017) proposed a time-space network that combines RMC production and vehicle dispatching, and the problem is

Referee List (DOI: 10.36253/fup_referee_list)
FUP Best Practice in Scholarly Publishing (DOI 10.36253/fup_best_practice)

optimized by a heuristic algorithm. Nonetheless, these studies overlook the specific properties of ERV, especially those related to the battery, such as charging and energy consumption.

To bridge this gap, our study proposes a scheduling optimization methodology for ERV dispatching. We first provide a detailed problem definition for ERV dispatching, incorporating the unique features of both EVs and RMC delivery tasks. The problem is then modeled as a Markov decision process (MDP) to capture the sequential logic of the RMC dispatching task. Finally, we propose an improved model-based reinforcement learning algorithm to solve the MDP problem. This algorithm, developed using a state-of-the-art Monte Carlo Tree Search (MCTS), is enhanced with a state-dependent action masking and a decaying searching strategy.

2. METHODOLOGY

2.1 ERV Operation Problem Definition

The operation of Electric Ready-mixed Vehicles (ERVs) comprises five components: a) The construction site and b) the ready-mixed concrete (RMC) plant, which are the locations where ERVs are prepared and RMC is poured. c) RMC mixer and d) charging station, which are machinery installed at the RMC plant for loading RMC and charging EVs respectively. Lastly, e) ERVs are the vehicles for dispatching RMC. For the purposes of this study, we assume pumps are pre-installed. The operational process of ERVs can be partitioned into three sections: the in-plant process (IP), the midway process (MP), and the on-site process (OP).

2.1.1 In-plant process

Prior to the delivery of the RMC batch to the construction site, it is imperative that an ERV is adequately prepared with the required RMC and sufficient battery power at the RMC plant. The specificities of two in-plant processes are as follows: **(1) IP1-RMC Production and Loading**: The RMC mixer produces and loads RMC onto ERVs according to the demands, which are typically determined based on the specific requirements of various construction sites. The following assumptions are made: a) RMC mixers can produce any type of required RMC, and the loading rate is set constant in this study for efficient validation. b) The RMC plant can load RMC onto multiple ERVs simultaneously, eliminating any queuing time for the loading (Z. Liu et al., 2017). c) Each ERV should be fully loaded unless it delivers the last batch of the target construction site, which can be smaller than an ERV's capacity. d) The plant owns various types of ERVs, and their loading capacities and operation costs are different (Z. Liu et al., 2017). **(2) IP2-Charging of ERVs**: When an EV's battery is less than its required degree, it will be recharged by charging stations. All the charging stations are installed only in the RMC plant, which is a regular practice in current ERV providers. a) We assume that the charging rates and costs are constant, but a basic cost is set to avoid frequent charging since launching the charging station is power-consuming. b) Multiple ERVs can be recharged simultaneously using multiple charging stations. c) ERVs have different battery capacities, and they can be recharged to a certain level between the existing status and the fully charged status.

2.1.2 Midway process

Following proper preparation, all ERVs should depart from the RMC plant to their corresponding construction sites. Two types of midway processes can be considered: **(1) MP1-Plant to Site**: a) A qualified pump is assumed to be installed on the construction site before the first arrival of ERV. b) To avoid unnecessary battery drainage while waiting with loaded RMC, it is assumed that the ERV will depart only if its arrival time is not earlier than the demand time of the target site. If the arrival time is earlier than the demand time, the ERV's RMC loading time, battery charging time, and travel time will be delayed. c) This study assumes that the ERV has the same traveling speed and battery consumption rate under the loading status. **(2) MP2-Site to Plant**: After unloading a batch of RMC at the construction site, the ERV returns to the plant and prepares for the next delivery batch of RMC. a) It is assumed all RMC have been unloaded, and the ERV is in an empty status. b) All ERVs return with a fixed traveling speed and energy consumption rate in the empty-load status.

2.1.3 On-site process

ERVs are assumed to arrive at the construction site not earlier than the demand time, which allows for the pouring task to commence promptly upon the ERV's arrival. a) Pourings are preferred to be consecutive, but delivery delays are also allowed in real applications. The construction sites claim a maximum for the delivery interval. b) Although static during the pouring process, ERVs remain operational, and a fixed battery consumption rate is assumed. c) This study supposes each construction requires only one ERV for pouring, and the pouring rate is fixed.

2.2 Modeling the ERV Operation Processes via an MDP Model

Based on the ERV operation problem definition, the maximum RMC batch of each construction site can be inferred by considering the minimum ERV capacity and site demands. This allows the ERV operation process to be modeled as a sequential decision-making problem, with the goal of optimizing the dispatch sequence of all the ERVs. During each dispatch, the ERV delivers a batch of RMC to a certain construction site with a certain battery level. Markov Decision Process (MDP) is a potent model-based method for sequential decision-making, which can be solved byiteratively evaluating the reward function for all potential states and actions until convergence to the optimal value (Zhang et al., 2020). Therefore, an MDP formulation is proposed for sequential coverage pattern analysis, represented by a four-element tuple (S, A, Ts,a, Ra). S is the state space, A is the action space, Ts,a is the state transition operator, and Ra is the reward function. These elements are discussed below.

2.2.1 State

$\boldsymbol{S}$ is the state space, $s \in S$ is the current state, which is a tuple of $2N_1 + 2N_2$ components. N_1 denotes the maximum number of ERVs owned by the RMC plant, while N_2 represents the maximum number of construction sites in demand. The MDP state comprises four parts, namely ERV's latest available time (LAT), ERV's battery status, the construction sites' latest demand time (LDT), and the quantities of undelivered RMC. The details of the MDP states are illustrated in Table 1.

Table 1 The definition of the MDP state.

State number	Meaning	Related component	Format
[1, N1]	The latest available time (LAT) of the 1st to the N1th ERVs.	ERV	Day-hour-minutes
[N1+1, 2N1]	The ERVs' battery states on their corresponding LAT.	ERV	kWh
[2N1+1, 2N1+N2]	The latest demand time (LDT) of the 1st to the N3th construction sites.	Construction site	Day-hour-minutes
[2N1+N2+1, 2N1+2N2]	The quantities of undelivered RCM for a certain construction site.	Construction site	m3

For clarification, we consider a scenario with two ERVs, two construction sites, and the state is (9:00, 9:30, 120, 60, 10:30, 12:00, 50, 25) (as shown in Fig. 1). It means that the first ERV is available after 9:00 with a 120-kWh battery, and the second ERV is available after 9:30 with a 60-kWh battery. The first construction site has a latest demand time (LDT) of 10:30 and requires 50 m^3 of RMC. The second construction site has an LDT of 12:00 and requires 25 m^3 of RMC.

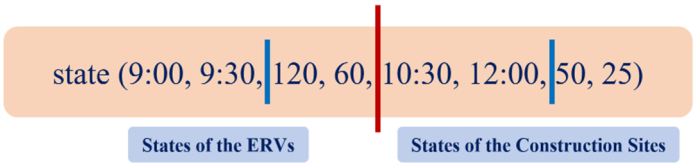

Fig. 1 An illustration of the MDP state

2.2.2 Action

$\boldsymbol{A}$ is the action space, where $a \in A$ is the action taken based on the current state. Each action is denoted by (e, c, b), where e is the serial number of the ERV, c is the serial number of each construction site, and b is the departure battery level. For the battery level, 0 means that the battery is kept unchanged, and its accuracy can be determined based on the user's computational capacity. Following the same scenario in section 3.3.1, we assume a battery accuracy is 5. Then, action (1, 2, 4) means the first ERV deliveries a batch of RMC to the second construction site with an 80-percentage battery, and action (2, 1, 0) means the second ERV is dispatched to the first construction site without charging.

2.2.3 Transition

$\boldsymbol{T_{s,a}}$ is the absolute transition that action a in state s at step t will lead to state s' at step $t+1$, as the transition is fully under control in this study. Apart from the state information, additional parameters are clarified for the state transition (Table 2).

Table 2 Fixed parameters for the state transition.

Fixed Parameters	Definition	Related element	Unit
l	The RMC loading rate of the mixer.	RMC mixer	m^3/min
p	The power of the charging station.	Charging station	kw
$g[e]$	The battery capacity of an electric vehicle	ERV	kWh
x	The battery accuracy	ERV	/
$m[e]$	The RMC capacity of the e^{th} ERV.	ERV	m^3
v_1	The vehicle speed of loading status.	ERV	km/hr
v_2	The vehicle speed of empty status.	ERV	km/hr
r_1	The battery consumption rate for the ERVs to travel under loading status.	ERV	%/km
r_2	The battery consumption rate for the ERVs to travel under empty status.	ERV	%/km
r_3	The battery consumption rate for ERVs to conduct the pouring task.	ERV	%/m^3
$d[c]$	The distance from the plant to the c^{th} construction site.	Construction site	km
w	The RMC pouring rate.	ERV	m^3/min

Firstly, the quantity of the RMC required by the target construction site ($s[2N_1 + N_2 + c]$) is compared with the capacity of the ERV ($m[e]$) to get the quantity of delivered RMC $RMC_{deliveried}$ (Eq.(1)). Subsequently, the loading time $t_{loading}$ can be obtained by Eq.(2). Given the action component b and the battery status ($s[N_1 + e]$), the charging time $t_{charging}$ can be calculated using Eq.(3). The departure time from the factory to the construction $t_{departure}$ can be determined by Eq.(4). Based on Eq.(5), the start time of the current dispatch t_{start} can be obtained by comparing the ERV's arrival time ($s[e] + t_{charging} + t_{loading} + t_{departure}$) with the construction site's LDT ($s[2N_1 + c]$). Further, Eqs.(6) and (7) are used to calculate the pouring time $t_{pouring}$ and return time t_{return}.

$$RMC_{deliveried} = \min(s[2N_1 + N_2 + c], m[e]) \quad (1)$$

$$t_{loading} = RMC_{deliveried}/l \quad (2)$$

$$t_{charging} = (\frac{b}{x} * g[e] - s[N_1 + e])/p \quad (3)$$

$$t_{departure} = d[c]/v_1 \quad (4)$$

$$t_{start} = \min\big(s[e] + t_{loading} + t_{charging} + t_{departure}, s[2N_1 + c]\big) \quad (5)$$

$$t_{pouring} = RMC_{deliveried}/w \quad (6)$$

$$t_{return} = d[c]/v_2 \quad (7)$$

The LAT of the current ERV $s[e]$ is updated by adding the pouring time and return time to the start time (Eq.(8)). The ERV's battery level $s[N_1 + e]$ is updated according to Eq.(9), where the second term is the battery consumption during traveling, and the third term is the battery consumption during the pouring task. The target construction site's LDT $s[2N_1 + c]$ is updated by adding the pouring time to the start time (Eq.(10)). Further, the required quantity of RMC $s[2N_1 + N_2 + c]$ can be updated by subtracting the quantity of delivered RMC from the target construction site's initial requirement, as shown in Eq.(11).

$$s[e] = t_{start} + t_{pour} + t_{return} \quad (8)$$

$$s[N_1 + e] = \left(\frac{b}{x} - d[c] * (r_1 + r_2) - RMC_{deliveried} * r_3\right) * g[e] \quad (9)$$

$$s[2N_1 + c] = t_{start} + t_{pour} \quad (10)$$

$$s[2N_1 + N_2 + c] = s[2N_1 + N_2 + c] - RMC_{deliveried} \quad (11)$$

2.2.4 Reward

R_a is the immediate reward of action a. In the RMC dispatch task, the objective of the RMC plant is to minimize the operational costs and adhere to the dispatch rules, such as avoiding exceeding the maximum pouring interval. Meanwhile, the construction sites aim to minimize the total delay for the pouring task. Therefore, the total

operation costs r_c and the dispatch delay r_d are selected as the two primary reward components. r_c can be calculated by Eq.(12). c_1 (\$/min) is the unit cost of the ERV operation, c_2 (\$) is the fixed cost of opening the charging station, which aims to avoid frequent charging, and c_3 (\$/kWh) is the unit price for ERV charging. The relevant costs of RMC production are not considered as the quantity of the RMC demand is fixed. According to Eq (13), r_d can be calculated by comparing ERV's arrival time with the construction site's LDT.

$$r_c = -c_1[e] * (t_{loading} + t_{charging} + t_{departure} + t_{pouring} + t_{return}) - c_2 * Boolean(t_{charging} > 0) - c_3 * (\frac{b}{x} * g[e] - s[N_1 + e]) \quad (12)$$

$$r_d = -\max(s[e] + t_{charging} + t_{loading} + t_{departure} - s[2N_1 + c], 0) \quad (13)$$

In addition, a significant negative reward r_p is generated if an invalid action is taken. Three types of invalid actions have been identified: 1) Actions that head to the construction site without any demand for RMC, as shown in Eq.(14); 2) Actions with a battery level below the current battery level or the minimum battery requirement (Eq.(15)). 3) Actions that result in a dispatch delay that exceeds the maximum interval δ (Eq.(16)). When the RMC demands of all the construction sites are fulfilled, a great positive reward r_f is generated. Finally, the total reward can be calculated by Eq.(17), where α_1 and α_2 are importance hyper-parameters. Apart from r_f, all other reward components are negative.

$$\textbf{\textit{InvalidAction}}\ \mathbf{1}: s[2N_1 + N_2 + c] > 0 \quad (14)$$

$$\textbf{\textit{InvalidAction}}\ \mathbf{2}: b > \max(s[N_1 + e], d[c] * (b_1[e] + b_2[e]) - RMC_{deliveried} * b_3[e]) \quad (15)$$

$$\textbf{\textit{InvalidAction}}\ \mathbf{3}: s[e] + t_{charging} + t_{loading} + t_{departure} - s[2N_1 + c] \leq \delta \quad (16)$$

$$r = r_f + \alpha_1 * r_c + \alpha_2 * r_d + r_p \quad (17)$$

2.3 Optimization Using PMD-MCTS

Many reinforcement learning methods have been developed to solve the MDP problem, including model-free algorithms and model-based algorithms (Sutton & Barto, 2018). Specifically, the model refers to the state transition function $T_{s,a}$ and the reward function R_a of the MDP problem. Compared with model-free methods, model-based reinforcement learning has the great potential to make RL algorithms more sample efficient (Wang et al., 2019). MCTS is a model-based RL algorithm that plans the best action at each time step (Browne et al., 2012). It is an effective heuristic search algorithm for solving episodic decision-making problems when the underlying search spaces are computationally expensive (B. Huang, Boularias, & Yu, 2022). However, MCTS relies on a large number of interactions with the environment emulator to construct the search trees for decision-making (Browne et al., 2012). To mitigate the high time complexity of classical MCTS, this section develops an improved MCTS algorithm， named parallel-masked-decaying MCTS (PMD-MCTS). Specifically, the state-of-the-art parallel MCTS algorithm, WU-UCT (A. Liu et al., 2018), is adopted as the fundamental model. It is further improved by incorporating a state-dependent action masking operation and a decaying search strategy. The details of the algorithm are introduced as follows.

2.3.1 Fundamentals of WU-UCT-based parallel Monte Carlo Tree Search

MCTS adopts a tree-search method that incrementally extends a search tree from the current environment state (Luo et al., 2022). Each node denotes a visited state, and each edge from this state denotes an action that can be taken at that state, leading to a landing node that denotes the state after the transition. Typically, MCTS performs four sequential steps repeatedly: selection, expansion, simulation, and backpropagation (Fig. 2 (a)). The selection step starts from the root node (current state) and recursively selects an existing child node according to a tree policy. The process ends when it reaches a leaf node or other termination conditions. One of the most commonly used node-selection policies is the Upper Confidence bound for Trees (UCT), and the UCT value a_s can be calculated by Eq.(18). Here, *C(s)* represents the child node set of the current node *s*; $V_{s'}$ is the average value estimation for a certain child node s', denoting the exploitation; The second term is the uncertainty of the value estimation, denoting the exploration. N_s and $N_{s'}$ denote the number of times that nodes s and s' have been visited, while β is the factor that controls the trade-off between exploitation and exploration. During the expansion state, a new child node is added to the selected node, and the value of the expanded node is estimated by performing a model-based simulation until termination. The simulation process follows a certain policy (e.g., random). Finally, backpropagation recursively updates the statistics $\{V_s, N_s\}$ from the expanded node to the root node along the selected path. According to Eq.(19), the visit count of each node is increased by 1. The latest value estimation of each node can be obtained by Eq.(20), where γ is the reward discount factor, and a_t is the action that turns the state s_t to the state s_{t+1}. It should be noted that the leaf node s_T obtains its value from the simulation step. Finally,

each node's average value estimation is updated according to Ep. (21).

$$a_s = \underset{s' \in C(s)}{\operatorname{argmax}} \left\{ V_{s'} + \beta \sqrt{\frac{2 \log N_s}{N_{s'}}} \right\} \tag{18}$$

$$N_{s_t} = N_{s_t} + 1 \tag{19}$$

$$V'_{s_t} = R(s_t, a_t) + \gamma V'_{s_{t+1}} \tag{20}$$

$$V_{s_t} = \left(\left(N_{s_t} - 1 \right) V_{s_t} + V'_{s_t} \right) / N_{s_t} \tag{21}$$

Parallelizing MCTS over multiple workers is an efficient method to improve the optimization speed. During the parallel computation, workers typically operate at different steps as the simulation and expansion processes are slow (Fig. 2 (b)). As a result, the update of statistics $\{V_s, N_s\}$ may become outdated for workers, and the statistics loss becomes inevitable. However, the latest N_s is available as soon as a worker initiates the computation since we only need to know if the node is selected. Therefore, the WU-UCT algorithm partially addresses the information loss by introducing another quantity O into the classical UCT (Eq.(22)), which counts the number of computations that have been initiated but not completed (light dashed blue lines in Fig. 2 (b)). The updated UCT effectively balances exploration-exploitation tradeoff by considering incomplete samples, and the node values can be updated according to Eqs.(23) and (24).

$$a_s = \underset{s' \in C(s)}{\operatorname{argmax}} \left\{ V_{s'} + \beta \sqrt{\frac{2 \log(N_s + O_s)}{N_{s'} + O_{s'}}} \right\} \tag{22}$$

$$\text{Incomplete update: } O_s = O_s + 1 \tag{23}$$

$$\text{Complete update: } O_s = O_s - 1, \quad N_s = N_s + 1 \tag{24}$$

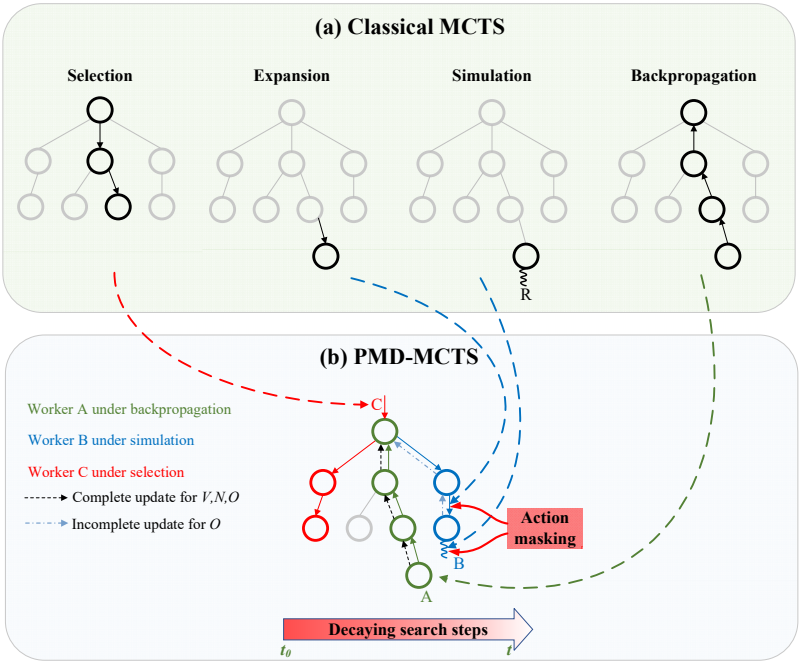

Fig. 2 The relationship between classical MCTS and PMD-MCTS

2.3.2 Action masking

The ERV operation involves complicated rules, and the valid action spaces usually vary under different states. Typically, RL algorithms sample the action from a space containing actions of all states and assign a significant negative reward for invalid actions. However, this kind of invalid action penalty is challenging to explore, particularly when the state is complicated, even for the very first reward (S. Huang & Ontañón, 2020). Hence, this section proposes a state-dependent action masking method to improve MCTS's efficiency (Fig. 2 (b)). First, a complete action space A_0 is generated, which contains all combinations of the ERV's serial number, each construction site's serial number, and the departure battery level. Then, invalid actions (Eqs.(14)-(16)) of A_0 are updated under each state (circles in Fig. 2 (b)), followed by invalid action masking. Specifically, $V_{s'}$ (Eq.(22)) of invalid actions are set as a large negative number M (e.g., M = -1 x 10^8) during the expansion stage. Consequently,

only valid actions will be expanded. During the simulation stage, actions are randomly selected from valid candidates, while invalid ones are ignored. In practical implementations, vectorization is adopted for speeding up the masking operations.

2.3.3 Decaying search strategy

During the implementation of MCTS, an initial MDP state is set as the start for optimization. The best child node of the current state is set as the output when the number of iterations is larger than a threshold value of N_δ. Then, the updated state becomes the new start, and the process iteratively continues until the MDP ends. Intuitively, the size of the Monte Carlo tree will decrease gradually, as the quantity of undelivered RMC decreases. Hence, the last stages of MDP may not require many iterations, and this section proposes a decaying threshold to ensure both optimization accuracy and efficiency. The decaying strategy is designed based on the remaining demand for RMC, as shown in Eq.(25). $N_{\delta 0}$ is the maximum number of iterations determined by the users, Q_{left} is the remaining required quantity of RMC, Q_{left} is the total RMC demand, and e is the Euler's number.

$$N_\delta = N_{\delta 0} * \frac{e^{\frac{Q_{demand}}{Q_{total}}}}{e} \tag{25}$$

3. VALIDATION

3.1 Scenario Setup

As the use of electric RMC vehicles is a relatively new solution in the construction industry, a customized dataset for this purpose is currently unavailable. Therefore, it is reasonable and acceptable to utilize data from previous RMC delivery studies to establish the proposed MDP model. Hence, we extracted the basic configurations of sites and RMC vehicles from the dataset of (Z. Liu et al., 2017). The dataset was determined based on a real case, including distances between the sites, RMC demands of the construction sites, RMC loading rate, ERVs' capacities, and relevant costs. We updated certain assumptions from (Z. Liu et al., 2017) in more detail. For example, we provided vehicle speeds for traveling time calculation and determined the battery-related factors based on actual reports (e.g., the charging rates). Table 3 describes the shared parameters, Table 4 indicates the information on the construction sites, and Table 5 shows ERV information. Two objectives are optimized: a) objective 1 aims to minimize the operation costs for the RMC plant, and b) objective 2 aims to minimize the dispatch delay for the construction sites.

Table 3 Information of the shared parameters.

Shared parameters	Value	Unit
RMC loading rate	2	min/m^3
Battery charging power	20	kw
Battery accuracy	5	/
Vehicle speed of loading status	40	km/hr
Vehicle speed of empty status	80	km/hr
Battery consumption rate for ERVs to travel under loading status	1	%/km
Battery consumption rate for ERVs to travel under empty status	0.8	%/km
Battery consumption rate for ERVs to conduct the pouring task	0.25	%/ m^3
RMC unloading rate	0.5	m^3/min
Cost of opening the charging station	5	$
Unit cost for ERV charging	2	$/kWh
Importance hyper-parameters α_1	1 for objective 1, 0 for objective 2	/
Importance hyper-parameters α_2	0 for objective 1, 1 for objective 2	/
Reward for an invalid action r_p	-1000	/
Reward for completing the task r_f	1000	/
Maximum pouring interval δ	90	min

Table 4 Information of the construction sites.

No.	RMC demand (m^3)	Distance (km)	Start time (hr:mm)
C1	6	6.2	8:30
C2	60	4.0	8:40
C3	26	5.5	9:30
C4	3	3.4	10:40
C5	64	12.0	11:20
C6	64	4.1	10:00
C7	24	6.6	15:10

Table 5 Information of the ERVs.

No.	1	2	3	4	5	6	7	8
RMC Capacity (m^3)	8	8	8	7	7	6	5	2
Unit cost ($/min)	1.3	1.3	1.3	1.2	1.2	1.1	1.0	0.8
Battery capacity (kWh)	160	160	160	160	160	120	100	50
Initial battery capacity (kWh)	80	80	80	80	80	60	50	25
Initial LAT (hr:mm)	6:00	6:00	6:00	6:00	6:00	6:00	6:00	6:00

3.2 Benchmark Setup

To validate the performance of our proposed PMD-MCTS algorithm under the given scenario setup, we compared it with three benchmarks, including GA-based optimization from (Z. Liu, Zhang, & Li, 2014), and two ablation studies. All algorithms were run ten times to minimize the impact of random errors. The two most common metrics, namely a) the average reward and b) the average computation speed, were used as the first two evaluation criteria. To test the stability of the algorithm, three additional metrics were selected, namely c) the success rate, d) the standard deviation (SD) of the average reward, and e) the SD of the average computation speed. Instead of terminating the MDP process when an invalid action occurs, we adopted a great negative number as a penalty and continued the MDP simulation. To avoid a negative battery state, the negative battery level was modified to the smaller one between the current battery status and the minimal battery requirement.

3.2.1 Genetic algorithm

Three-layer chromosome: The chromosome structure was designed based on the concepts of (Karakatič, 2021; Z. Liu et al., 2014). As described in (Z. Liu et al., 2014), the maximum number of vehicles to be dispatched is fixed, which is set as the chromosome length. The chromosome of (Z. Liu et al., 2014) has three layers: a) sequence of construction site ID, b) sequence of the accumulative number of vehicles to the construction site, and c) sequence of vehicle ID. The second layer was removed as it can be inferred from the first layer. In addition, we added a layer for battery level according to (Karakatič, 2021), and used the same battery definition as our PMD-MCTS method. An illustration of the chromosome is shown in Fig. **3**.

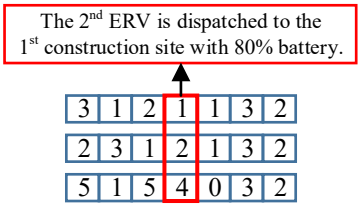

Fig. 3 An illustration of the GA chromosome

Selection: The chromosome represents a set of sequential MDP actions that can be input into the MDP model to obtain the accumulative reward (fitness). It should be noted that action will not be taken if the target construction site has been satisfied.

Crossover: This study adopted one-point crossover, but the crossover operation may change the maximum number of vehicles required by each site. Hence, the probability mapping method of (Z. Liu et al., 2014) was adopted for the first layer crossover, as shown in Fig. **4**. Specifically, each gene in the first layer has a mapping probability, and the crossover is conducted on the probability layer. The new chromosome is generated by mapping the probabilities to a basic chromosome in descending order, and the basic chromosome can be user-defined ([1,1,1,2,2,3,3] in Fig. **4**). The crossovers of the second and third layers are conducted directly. We conduct the

crossover layer by layer, which can generate six children during one crossover.

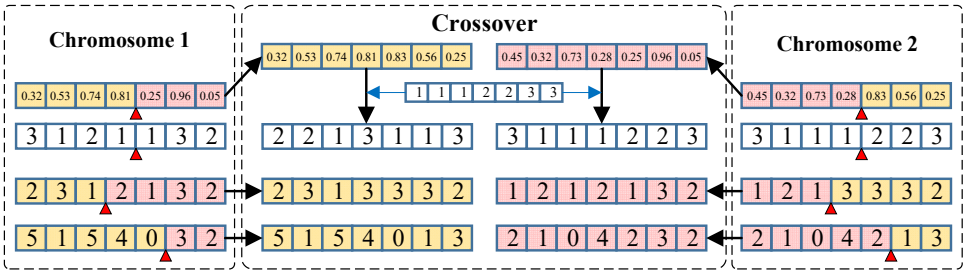

Fig. 4 The crossover of two chromosomes

Mutation: One-point mutation is adopted, as shown in Fig. **5**. Similar to the crossover operation, the mutation of the first layer is realized by probability mapping, while the genes in the other two layers are mutated according to their ranges.

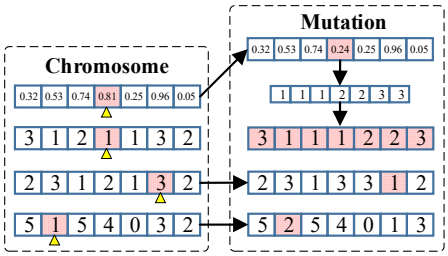

Fig. 5 The mutation of the chromosome

Table 6 Hyperparameters of genetic algorithm

Hyperparameters	Value
Population size	20
Parent number	3
Mutation rate	0.3
Maximum generation number	200000

3.2.2 Ablation studies

We have made two improvements based on WU-UCT-based MCTS. To evaluate the effectiveness of these improvements, we conducted two ablation studies: a) PD-MCTS, which is PMD-MCTS without action masking, and b) PM-MCTS, which is PMD-MCTS without decaying search strategy. The hyperparameters used in the MCTS algorithms are listed in Table 7.

Table 7 Hyperparameters of MCTS algorithms

Hyperparameters	Value
Number of expansion workers	8
Number of simulation workers	16
Maximum search step ($N_{\delta 0}$)	3000
Maximum search depth	100
Maximum search width	200
Discount factor	0.9
Expansion policy	Random

3.3 Results

Experiments for PMD-MCTS and four benchmark algorithms were conducted on the designed scenario. The entire procedure was executed on a laptop with the specification of Intel i9-10980H 3.10GHz CPU and 32GB RAM. The the scheduling performance of each algorithm is shown in Table 9 and Table 8.

.

Table 8 Scheuling performance of each algorithm in objective_1 (minimizing the costs)

	Average reward	SD of average reward	Success rate (%)	Average computation time (s)	SD of average computation time
GA	-8313.1(-6494.3*)	1283.1	0	102.0	**1.7**
PD-MCTS	-6950.0 (-5421.5*)	1040.6	0	320.8	26.3
PM-MCTS	-3095.2 (-3004.5*)	**120.4**	**100**	385.6	40.1
PMD-MCTS	**-3064.0 (-2862.4*)**	123.0	**100**	260.0	19.9

Table 9 Scheuling performance of each algorithm in objective_2 (minimizing the delay)

	Average reward	SD of average reward	Success rate (%)	Average computation time (s)	SD of average computation time
GA	-837.9 (479.4*)	1076.5	30	93.27	2.0
PD-MCTS	-484.1 (182.8*)	503.3	**100**	497.8	78.1
PM-MCTS	969.0 (1000*)	18.2	**100**	292.4	10.4
PMD-MCTS	**991.6 (1000*)**	**6.7**	**100**	203.0	**1.6**

* indicates the optimal performance

The empirical results indicate that our PMD-MCTS algorithm demonstrates superior performance, achieving the highest rewards across both objectives. Specifically, the average reward of the PMD-MCTS in objective 1 is 3064.0, which translates to an average cost of $4064.0. In objective 2, the average reward of the PMD-MCTS is 991.6, representing an average delay of 8.4 minutes, and the most optimal solution can eliminate any delay entirely. Furthermore, our findings suggest that only algorithms implementing action masking (i.e., PMD-MCTS and PM-MCTS) can consistently ensure a feasible solution for both objectives. These two algorithms also display the smallest standard deviation of average reward, indicating their superior stability. Although PM-MCTS exhibits a performance similar to PMD-MCTS in terms of reward and success rate, the PMD-MCTS requires 30% less computational time than PM-MCTS.

4. DISCUSSIONS

This study's outcomes substantiate the effectiveness of our proposed scheduling optimization approach for managing ERV operations. This methodology mainly contributes to the field in three ways.

Firstly, this study addresses an existing gap in on-road Commercial Electric Vehicle (CEV) research. We are pioneers in examining CEVs, particularly on-road CEVs, marking a significant stride towards sustainable advancement in the construction sector. By incorporating the demands of Ready-Mixed Concrete (RMC) dispatching and Electric Vehicles (EVs), we have holistically examined the characteristics of ERVs. This problem definition can potentially be extrapolated to other CEV studies in the future. Secondly, we have crafted a novel formulation for the RMC delivery problem, utilizing the Markov Decision Process (MDP) based on the temporal dynamics of the RMC delivery process. Compared to its predecessors, the MDP formulation is a more rational choice as it facilitates sequence decision-making. This approach prevents invalid decisions at each stage and ensures the decision-making process is far-sighted, considering all decisions in a comprehensive manner. Lastly, we introduced an enhanced Monte Carlo Tree Search (MCTS) algorithm, named PMD-MCTS, to optimize the ERV operation process. When compared with four benchmark algorithms, it proved to be the most effective. Two key advantages of the PMD-MCTS were identified: Both PMD-MCTS and PM-MCTS displayed superior performance in terms of average reward and success rate, outperforming the Genetic Algorithm (GA) by employing the MCTS optimization strategy. The GA algorithm fails to ensure a feasible solution for both objectives, owing to its limitations in managing sequential requirements. PMD-MCTS surpassed PM-MCTS on computational speed. Our PMD-MCTS saves over 30% of the computational time required by PM-MCTS, without compromising on accuracy, by implementing a decaying strategy.

5. CONCLUSION

In the face of pressing concerns over carbon emissions, the construction industry can expect to see an influx of more sustainable technologies. Electric Ready-mixed Vehicles (ERVs) are a promising technology geared towards enhancing the sustainability of the construction industry. However, the interdisciplinary nature of ERVs has led to a considerable gap in this field. This study addressed this gap by proposing a scheduling optimization methodology for ERV dispatching. It introduces a systematic problem definition for the ERV operation, which integratively considers the properties of both EVs and RMC delivery tasks. Moreover, the ERV operation process is modelled as an MDP problem, thereby breaking down the entire process into sequential sub-processes. The proposed PMD-MCTS algorithm, equipped with parallel computing, invalid action masking, and decaying searching capability, has been validated through a meticulously designed experiment. This study, thus, provides a comprehensive evaluation of ERV operations and offers a solid foundation for future research in this domain.

REFERENCES

Bogachkova, L. Y., Guryanova, L. S., & Usacheva, N. Y. (2022). Decarbonization Trends in the Largest Post-soviet Countries and the Specifics of Their Inclusion in the Global Climate Agenda. *New Technology for Inclusive and Sustainable Growth: Perception, Challenges and Opportunities*, 77-88. doi:https://doi.org/10.1007/978-981-16-9804-0_7

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., . . . Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games, 4*(1), 1-43. doi:https://doi.org/10.1109/TCIAIG.2012.2186810

Gan, V. J., Chan, C. M., Tse, K., Lo, I. M., & Cheng, J. C. (2017). A comparative analysis of embodied carbon in high-rise buildings regarding different design parameters. *Journal of Cleaner Production, 161*, 663-675. doi:https://doi.org/10.1016/j.jclepro.2017.05.156

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics, 4*(2), 100-107. doi:https://doi.org/10.1109/TSSC.1968.300136

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780. doi:https://doi.org/10.1162/neco.1997.9.8.1735

Huang, B., Boularias, A., & Yu, J. (2022). *Parallel Monte Carlo Tree Search with Batched Rigid-body Simulations for Speeding up Long-Horizon Episodic Robot Planning.* Paper presented at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Huang, S., & Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*. doi:https://doi.org/10.32473/flairs.v35i.130584

Karakatič, S. (2021). Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. *Expert Systems with Applications, 164*, 114039. doi:https://doi.org/10.1016/j.eswa.2020.114039

Lin, P.-C., Wang, J., Huang, S.-H., & Wang, Y.-T. (2010). Dispatching ready mixed concrete trucks under demand postponement and weight limit regulation. *Automation in Construction, 19*(6), 798-807. doi:https://doi.org/10.1016/j.autcon.2010.05.002

Lin, T., Lin, Y., Ren, H., Chen, H., Chen, Q., & Li, Z. (2020). Development and key technologies of pure electric construction machinery. *Renewable and Sustainable Energy Reviews, 132*, 110080. doi:https://doi.org/10.1016/j.rser.2020.110080

Liu, A., Chen, J., Yu, M., Zhai, Y., Zhou, X., & Liu, J. (2018). Watch the unobserved: A simple approach to parallelizing monte carlo tree search. *arXiv preprint arXiv:1810.11755*. doi:https://doi.org/10.48550/arXiv.1810.11755

Liu, Z., Zhang, Y., & Li, M. (2014). Integrated scheduling of ready-mixed concrete production and delivery. *Automation in Construction, 48*, 31-43. doi:https://doi.org/10.1016/j.autcon.2014.08.004

Liu, Z., Zhang, Y., Yu, M., & Zhou, X. (2017). Heuristic algorithm for ready-mixed concrete plant scheduling with multiple mixers. *Automation in Construction, 84*, 1-13. doi:https://doi.org/10.1016/j.autcon.2017.08.013

Luo, F.-M., Xu, T., Lai, H., Chen, X.-H., Zhang, W., & Yu, Y. (2022). A survey on model-based reinforcement learning. *arXiv preprint arXiv:2206.09328*. doi:https://doi.org/10.48550/arXiv.2206.09328

Olanrewaju, O. I., Edwards, D. J., & Chileshe, N. (2020). Estimating on-site emissions during ready mixed concrete (RMC) delivery: a methodology. *Case Studies in Construction Materials, 13*, e00439. doi:https://doi.org/10.1016/j.cscm.2020.e00439

Palaniappan, S., Bashford, H., Li, K., Fafitis, A., & Stecker, L. (2009). Carbon emissions based on transportation for post-tensioned slab foundation construction: A production home building study in the greater phoenix arizona area. *International journal of construction education and research, 5*(4), 236-260. doi:https://doi.org/10.1080/15578770903355533

Sinha, R. K., & Chaturvedi, N. D. (2019). A review on carbon emission reduction in industries and planning emission limits. *Renewable and Sustainable Energy Reviews, 114*, 109304. doi:https://doi.org/10.1016/j.rser.2019.109304

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*: MIT press.

Tan, S., Yang, J., Zhao, X., Hai, T., & Zhang, W. (2018). Gear ratio optimization of a multi-speed transmission for electric dump truck operating on the structure route. *Energies, 11*(6), 1324. doi:https://doi.org/10.3390/en11061324

Tong, Z., Jiang, Y., Tong, S., Zhang, Q., & Wu, J. (2023). Hybrid drivetrain with dual energy regeneration and collaborative control of driving and lifting for construction machinery. *Automation in Construction, 150*, 104806. doi:https://doi.org/10.1016/j.autcon.2023.104806

Turan, B., Pedarsani, R., & Alizadeh, M. (2020). Dynamic pricing and fleet management for electric autonomous mobility on demand systems. *Transportation Research Part C: Emerging Technologies, 121*, 102829. doi:https://doi.org/10.1016/j.trc.2020.102829

Volvo Trucks delivers the first heavy-duty electric concrete mixer truck to CEMEX. (2023).

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., . . . Ba, J. (2019). Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*. doi:https://doi.org/10.48550/arXiv.1907.02057

Zhang, X., Zhang, J., Liu, Z., Cui, Q., Tao, X., & Wang, S. (2020). MDP-based task offloading for vehicular edge computing under certain and uncertain transition probabilities. *IEEE Transactions on Vehicular Technology, 69*(3), 3296-3309. doi:https://doi.org/10.1109/TVT.2020.2965159