This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410

Multi-Agent Deep Reinforcement Learning based Scheduling Approach for Mobile Charging in Internet of Electric Vehicles

Linfeng Liu, Zhuo Huang, and Jia Xu,

***Abstract*—Mobile charging stations (MCSs) have become an indispensable complement of fixed charging stations. In the regions where fixed charging stations are sparsely deployed or even absent, the main concern is that how to properly schedule MCSs to charge the electric vehicles with insufficient electricity (EVCs). In this paper, we focus on the scheduling of idle MCSs and pending EVCs. To increase the charging revenue of MCSs and enhance the proportion of successfully charged EVCs, we schedule idle MCSs to proactively track some EVCs with potential charging demand, and schedule pending EVCs to approach some busy MCSs for potential charging opportunities. To this end, a Scheduling Approach based on Multi-Agent Deep Reinforcement Learning (SA-MADRL) is proposed to train the scheduling models for agents (idle MCSs and pending EVCs). In SA-MADRL, the agents obtain the local observations to make the scheduling decisions. Both idle MCSs and pending EVCs can independently make the scheduling decisions, and thus SA-MADRL can realize the fully distributed scheduling and has a good scalability. Extensive simulations and comparisons demonstrate the performance superiority of SA-MADRL, i.e., the charging revenue of MCSs can be significantly increased, and the proportion of successfully charged EVCs can be effectively enhanced.**

***Index Terms*—Internet of electric vehicles; mobile charging station; multi-agent deep reinforcement learning; scheduling strategy.**

I. Introduction

In recent years, due to the environmental concerns such as serious atmospheric pollution and rapid depletion of fossil resource, electric vehicles (EVs) have increasingly become the preferred mode of transportation, gradually replacing the conventional fuel-powered vehicles [1], [2]. However, owing to the limited battery capacity of EVs, EV drivers typically pay special attention to the residual electricity during their travels, easily leading to the effect of range anxiety.

Currently, fixed charging stations (FCSs) are the primary infrastructure for charging EVs. Nevertheless, some issues (such as the sparse/uneven distribution of FCSs and the long charging queues at FCSs) significantly impede the prompt charging of EVs [3]. As a complementary charging solution to FCSs, mobile charging stations (MCSs) can provide EVs with increased flexibility and convenience in the charging services by dynamically optimizing the distribution of charging capability [4] (employing some reasonable scheduling strategies). Together, EVs and the charging infrastructure (FCSs and MCSs), constitute the fundamental components in the Internet of Electric Vehicles (IoEV) [5] (Fig. 1), where EVs can communicate with the cloud server, charging infrastructure, and other vehicles to facilitate the real-time information exchanges and remote controls.

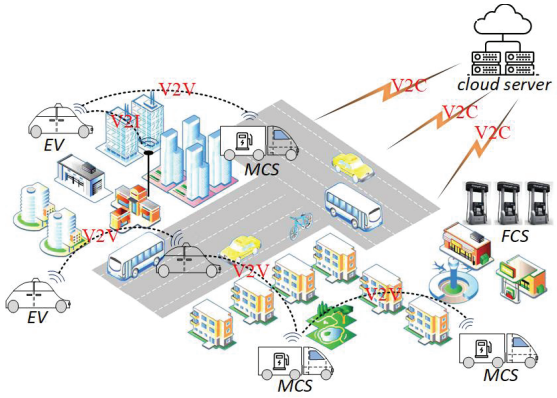


Fig. 1: IoEV with EVs and MCSs.

Some EVs do not have sufficient residual electricity to complete the planned travels, and thus require the charging to increase the mileage endurance. In this paper, when the residual electricity of these EVs drops below a low battery state, they are termed pending EVCs (electric vehicles to be charged); When the residual electricity of these EVs is higher than the low battery state, they are termed quasi EVCs.

In this paper, we focus on the mobile charging problem in the regions where FCSs are sparsely deployed or even absent, and the main concern is that how to properly schedule MCSs to charge EVCs. Each pending EVC will make a charging request to nearby idle MCSs. However, there could be no idle MCSs nearby which can receive this charging request, forcing the pending EVC to continue the travel and await the charging responses from idle MCSs in future. In addition, there are also some idle MCSs which do not reach any charging consensuses with EVCs in the following situations: (i) They do not receive any charging requests from pending EVCs; (ii) They are not selected by pending EVCs despite receiving their charging requests.

L. Liu, Z. Huang, and J. Xu are with the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, and also with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing 210023, China.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410

Therefore, we specially investigate the scheduling of idle MCSs and the scheduling of pending EVCs which have not received any charging responses from MCSs.

Intuitively, by scheduling the idle MCSs to proactively track the quasi EVCs, prompt charging services (quick charging responses) can be provided once the charging requests are made by the pending EVCs (when quasi EVCs turn into pending EVCs). Besides, proper routes should be recommended to the pending EVCs, directing them to seek the potential charging opportunities provided by busy MCSs in future. Specifically, for the pending EVCs, if there are no idle MCSs nearby, and there are some busy MCSs about to complete their current charging tasks, the pending EVCs could move close to these busy MCSs and wait for the future charging opportunities. By the above mechanisms, the charging revenue of MCSs can be increased, and the proportion of successfully charged EVCs can be enhanced.

In our work, quasi EVCs are configured to periodically broadcast the positions and electricity shortage within their communication range. The electricity shortage of a quasi EVC denotes the insufficient electricity for the quasi EVC to reach the intended destination. The idle MCSs could track some quasi EVCs when they do not receive any charging requests from the pending EVCs. When a quasi EVC turns into a pending EVC and makes a charging request, each idle MCS receiving the charging request will send back a charging response and negotiate with the pending EVC about the charging position, and each busy MCS provides the current position and the status of ongoing charging task, serving as the basis for the scheduling of the pending EVC if it does not receive any charging responses.

An example is given in Fig. 2, each pending EVC broadcasts the charging request within its communication range. EVC1 and MCS1 are successfully matched, proceeding to the negotiated charging position. However, EVC2 lacks access to the charging services of idle MCSs, hence it progressively approaches MCS2 which currently undertakes a charging task, waiting for the future charging opportunity. MCS3 (in idle state) does not receive any charging requests, and there are two nearby quasi EVCs (quasi EVC1 and quasi EVC2). MCS3 tracks quasi EVC1 and quasi EVC2 by moving towards a position with a high possibility of receiving the charging requests from quasi EVC1 and/or quasi EVC2.

Moreover, as quasi EVCs consistently attract the surrounding idle MCSs, and there may also arise an aggregation of multiple pending EVCs around some busy MCSs that are about to complete the charging tasks, leading to a local imbalance between charging supply and charging demand, i.e., many idle MCSs could track the same quasi EVCs, and many pending EVCs could move close to the same busy MCSs. Thereby, the cooperation and competition among MCSs and EVCs should be carefully considered to improve the mobile charging efficiency.

Motivated by the above facts, to model and investigate the cooperation and competition among MCSs and EVCs, this paper proposes a Scheduling Approach based on Multi-Agent Deep Reinforcement Learning (SA-MADRL). In particular, SA-MADRL trains the scheduling strategies for two different categories of agents: idle MCSs and pending EVCs, and the agents obtain the local observations and action intentions of nearby agents through vehicle to vehicle (V2V) communications to make the action decisions.

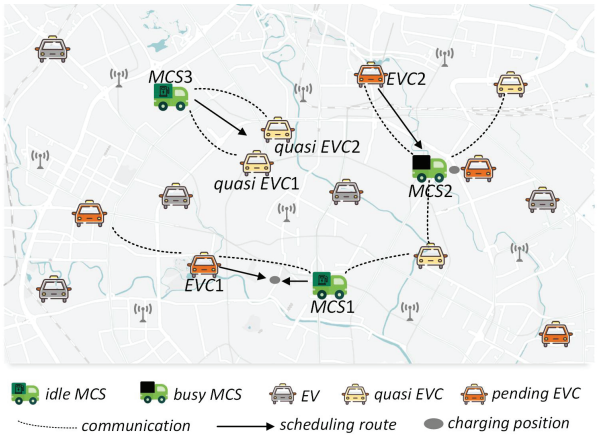


Fig. 2: An example regarding the scheduling of idle MCSs and pending EVCs.

Specifically, the decision-making processes of the two categories of agents are as follows: (i) Idle MCSs continuously monitor the charging requests from pending EVCs and the potential charging demand from quasi EVCs, and exchange the status information and action intentions [1] with other nearby idle MCSs. The information obtained by each agent (idle MCS) is input to an MCS action-value network to generate the scores for the scheduling points (the positions that the agent can be scheduled to). Then, each agent is dispatched to the scheduling point with the highest score in the local observation. (ii) Likewise, with regard to each pending EVC, it is necessary to collect the information of busy MCSs and other pending EVCs in the local observation, and then input the information into an EVC action-value network to generate the scores and obtain the optimal scheduling point for the pending EVC.

In addition, note that our proposed SA-MADRL can be distributed to each MCS or each EVC in the real world after being trained offline using a simulated environment on a cloud server (Fig. 3). Both idle MCSs and pending EVCs can independently make action decisions via their action-value networks, respectively, i.e., SA-MADRL can realize the fully distributed scheduling and thus has a good scalability.

The contributions of this work are summarized as follows: (i) We formulate the mobile charging scheduling problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), and schedule both idle MCSs and pending EVCs. (ii) A mean-field multi-agent deep reinforcement learning-based approach is proposed to deal with the scheduling problem of idle MCSs and pending

[1]The action intention is one of the available scheduling points in the local observation of an agent.

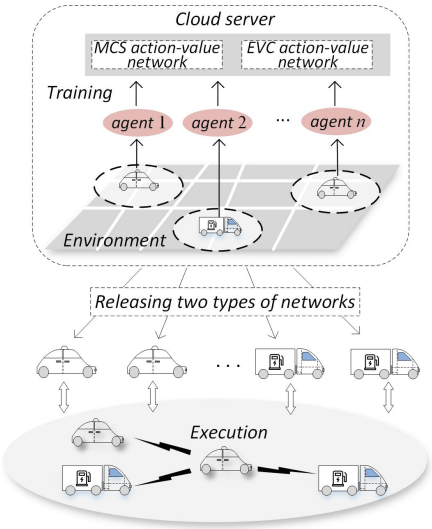


Fig. 3: Framework of scheduling idle MCSs and pending EVCs.

EVCs. This approach enables agents to achieve global interaction within the constraints of V2V communication range and coordinates the cooperation and competition among agents effectively. (*iii*) An offline simulated environment is implemented to train two multi-agent deep reinforcement learning models. Deploying the trained models to MCSs and EVCs enables the independent decision-making, and enhances the scheduling efficiency and system scalability.

The rest of the paper is organized as follows: Section II summarizes some related works. Section III elaborates on the system model for the scheduling problem of idle MCSs and pending EVCs. Section IV proposes a multi-agent deep reinforcement learning model for solving the scheduling problem. Section V describes the details of our proposed SA-MADRL. Section VI presents extensive simulation results to evaluate the performance of SA-MADRL. Section VII concludes this paper. The section of appendix provides some theoretical analysis of SA-MADRL.

II. Related Work

A. Charging with Fixed Charging Stations

FCSs serve as the primary solution for charging EVs and have been extensively deployed in urban regions. Numerous studies have examined the optimal arrangement of FCSs to meet the charging requirements and improve the overall profitability of FCSs. For example, [6] analyzes the charging demand of different regions in a city, a multi-objective optimization problem that minimizes the energy loss, voltage deviation, and land cost is established. By jointly considering the behavioral uncertainties of EVs and the influence factors such as state of charge (SoC), [7] introduces a Mixed-Integer Linear Programming (MILP) model to determine the optimal locations of FCSs. [8] proposes a capacitated deviation flow refueling location model, which concurrently addresses the optimal placement and capacity of FCSs. This model takes into account the uncertainty of renewable energy to fulfill the charging requirements of EVs while minimizing the adverse impacts on the power grid.

Moreover, many studies recommend the best charging strategy for EVs. In [9], a method for managing the charging and discharging of EVs is proposed to minimize the charging cost of EVs, enhance the battery longevity, alleviate the grid stress, and meet the charging requirements of EVs. The charging revenue of FCSs is strongly related with their business operations, such as the dynamic pricing mechanisms which include two categories: traditional optimization methods and reinforcement learning based methods. Traditional optimization methods such as linear optimization [10], genetic algorithm [11], and game theory [12] have been used to address the dynamic pricing for FCSs. However, for complex dynamic systems, these methods are computationally expensive and cannot make real-time decisions. The reinforcement learning based methods, such as [13], [14], [15], typically apply the model-free reinforcement learning frameworks to yield the dynamic pricing strategies and increase the long-term revenue of FCSs. Furthermore, considering the grid-connected charging stations where FCSs integrate the photovoltaic power generation and energy storage batteries, [16] provides an approximate dynamic programming feedback-based optimization method to solve the optimal control problem of EV fleet charging, and this method has the adaptation to dynamic changes in electricity price.

B. Charging with Mobile Charging Stations

To offer increased flexibility in charging services, MCSs serve as a valuable complementary solution to FCSs. There are two primary challenges in the scheduling problem of MCSs: how to balance the charging supply and charging demand, and how to design the reasonable scheduling routes of MCSs.

To address the first challenge, as introduced in [17] and [18], two effective methods have been introduced to consider the charging demand in peak hours. Specially, in [17], a reinforcement learning (RL) method based on policy evaluation is used to determine the scheduling positions of MCSs, and this method is combined with a fixed order dispatching algorithm. Nevertheless, the single-agent RL cannot handle the cooperation among multiple MCSs. In [18], a coordinated system is proposed to manage the charging plans of EVs to reduce the travel time and meet the charging demand of EVs during peak hours by using both FCSs and MCSs. In order to realize the efficient matching between EVs and MCSs, [19] formulates the decision problem of charging locations, monetary cost, and reward as a joint optimization problem. This work gives a software-defined EV-to-EV (MCS is an EV equipped with mobile charging equipment) charging framework which can make the preferable pair decisions. However, the above works ignore the routing issue in the scheduling process of MCSs, and note that the reasonable scheduling routes can enable MCSs to meet more charging demand.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410

TABLE I: Comparisons of some related works (the symbol "$\surd$" indicates that the issue is considered, and the symbol "$\times$" indicates that the issue is not considered.)

Method	Region scheduling of MCSs	Route scheduling of MCSs	Scheduling of EVs	Charging revenue of MCSs	Charging experience of EVs	Distributed scheduling	Cooperation and competition among MCSs and EVs
[17]	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$
[18]	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\times$
[19]	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\surd$
[20]	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$
[21]	$\times$	$\surd$	$\times$	$\times$	$\surd$	$\times$	$\times$
[22]	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\times$
[23]	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\surd$
SA-MADRL (our work)	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

With regard to the scheduling routes of MCSs, [20] predicts the potential charging positions for idle MCSs based on the historical trajectories and charging demand, thus increasing the proportion of successfully charged EVs. In addition, the use of federated learning allows each MCS to train the model locally, thereby shortening the training time. [21] studies a RL-based framework for the multi-objective scheduling problem to maximize the charging efficiency of EVs, by optimizing the charging sequence and actual charging energy. In [22], a charging planning and online operating system is proposed, the scheduling of MCSs is modeled as a dynamic vehicular routing problem, and an agent-based optimization algorithm is adopted to determine the number, battery capacity, and locations of MCSs. The online centralized scheduling system can well cope with the dynamic changes in real situations, but it is difficult to achieve good performance in large-scale charging scenarios. [23] proposes a bi-level optimization framework to maximize the charging revenue of MCSs. The upper level mainly solves the routing problem for MCSs, and the lower level mainly solves the scheduling problem of the energy service stations. However, this model does not take into account the charging experience of EV drivers.

The main differences of these related works are listed in TABLE I. As shown in TABLE I, the previous works focus on the scheduling of MCSs by considering the charging demand of EVs or taking MCSs as the temporary FCSs. However, these works typically ignore the willingness of EVs (EV drivers) of proactively seeking the potential charging opportunities when there are no available idle MCSs around them. Our work aims to schedule idle MCSs and pending EVCs simultaneously, and provide them with proper scheduling routes that take into account the charging revenue of MCSs and the charging experience of EVs. Moreover, considering the large-scale charging scenarios with many MCSs and EVs, our work also places emphasis on the cooperation and competition among MCSs and EVs, and realizes a fully distributed scheduling.

C. Multi-Agent Reinforcement Learning Methods

Reinforcement learning methods have been widely used in many applications such as traffic control and vehicle scheduling, often involving the multi-agent environments. Note that employing the single-agent reinforcement learning method directly in multi-agent environments could be confronted with many difficulties. The synchronous strategy updates on agents construct a non-stationary environment [24] where the convergence is seriously impeded, and the coordination of cooperation and competition among agents is significantly hindered.

Some Multi-Agent Reinforcement Learning (MARL) methods adopt the framework of Centralized Training with Decentralized Execution (CTDE) [25]. Specifically, there are two types of methods under the CTDE framework: The first type is the multi-agent policy gradient, where each agent has a decentralized actor network for the policy approximation and a centralized critic network for the action value evaluation [26], [27], respectively. The other type is the value decomposition method [28], which is based on the assumption that the sum of local maximum value of a single agent's action is equal to the global maximum value of the joint action.

In the mobile charging scenario, with regard to the scheduling of idle MCSs and pending EVCs, a vast state space will be generated due to the large number of agents and the intricate road network. Furthermore, the dynamics in the states of MCSs and EVs could result in a continuously varying number of agents, making it extremely difficult to conduct the centralized training based on the global states and joint actions. Inspired by Mean-Field MARL (MFMARL) [29], each agent makes the optimal decision based on the local observation and the average effect of other nearby agents. From the perspective of a single agent, this approach effectively reduces the non-stationarity in the multi-agent environments. Considering the wireless communication capability of MCSs and EVs, each agent can communicate with other nearby agents falling into its communication range, and then can exploit the local observations and action intentions of the nearby agents to make the scheduling decision.

III. System Model and Problem Formulation

The road network in the 2D plane is represented by $\mathcal{R}$, and the charging positions are selected from a set of charging parking lots $\mathcal{P}$ ($\mathcal{P} \in \mathcal{R}$). There exist N EVs and M MCSs, denoted by $\mathcal{N} = \{v_1, \cdots, v_N\}$ and $\mathcal{M} =$

$\{\phi_1, \cdots, \phi_M\}$, respectively. The following definitions are introduced to depict the scheduling problem of idle MCSs and pending EVCs.

TABLE II shows the list of main notations.

TABLE II: Main notations

Parameter	Description
o_i, ℓ_i, $d(o_i, \ell_i)$	Departure position of EV v_i, destination of EV v_i, travel distance from o_i to ℓ_i
$e(v_i)^{(t)}$, $e(\phi_j)^{(t)}$	Residual electricity of EV v_i, MCS ϕ_j at the t-th time slot
$p(v_i)^{(t)}$, $p(\phi_j)^{(t)}$	Position of EV v_i, MCS ϕ_j at the t-th time slot
$ms(v_i)$, $ms(\phi_j)$	Travel speed of EV v_i, MCS ϕ_j
$St(v_i)$	Electricity shortage of EVC v_i
R_c	Communication range of EVs and MCSs
$Dtr(v_i, \phi_j)$	Detour distance of EVC v_i charged by MCS ϕ_j
$T_W(v_i, \phi_j)$	Delay of EVC v_i waiting for MCS ϕ_j
$\xi(v_i)$	Charging delay of EVC v_i
$Rev(\phi_j)$	Charging revenue earned by MCS ϕ_j
c	Electricity consumption for travelling through a unit distance
ω	Charging speed
ϱ_f, ϱ_0	Unit price of electricity transferred from MCSs to EVCs, unit price of electricity purchased from power grid
$\mathcal{L}$	Low battery state

A. Electric Vehicles

Suppose each EV has the same communication range denoted by R_c. For an EV v_i travels from the departure position o_i to the destination ℓ_i with initial battery capacity of $e(v_i)^{(0)}$. The travel distance from o_i to ℓ_i is marked by $d(o_i, \ell_i)$, and v_i travels at a speed of $ms(v_i)$, consuming c unit of electricity for travelling through a unit distance.

For v_i, if the total electricity required for the travel $c \cdot d(o_i, \ell_i)$ exceeds the initial battery capacity $e(v_i)^{(0)}$, then v_i is a quasi EVC, and the electricity shortage of v_i is written as $St(v_i) = c \cdot d(o_i, \ell_i) - e(v_i)^{(0)}$. When v_i has started the travel for $(t-1)$ time slots, the residual electricity of v_i (at the t-th time slot) is marked as $e(v_i)^{(t)}$. If $e(v_i)^{(t)}$ is smaller than a low battery state $\mathcal{L}$ at the t-th time slot, v_i becomes a pending EVC and makes a charging request within the communication range R_c.

After making the charging request, if v_i receives one or more charging responses from nearby idle MCSs, and then v_i selects the idle MCS whose charging position can produce the shortest detour distance for v_i. If v_i does not receive any charging responses, and then v_i is still taken as a pending EVC, and the scheduling of v_i is conducted by our proposed SA-MADRL (introduced in Section V).

B. Mobile Charging Stations

Similar to EVs, each MCS has the same communication range R_c and consumes c unit of electricity for travelling through a unit distance. The travel speed of an MCS ϕ_j is denoted by $ms(\phi_j)$. If ϕ_j undertakes a charging task at the t-th time slot (e.g., ϕ_j is moving towards a charging position to charge another EVC $v_{i'}$, or ϕ_j is charging $v_{i'}$ at the charging position), and then ϕ_j is taken as a busy MCS; Otherwise, ϕ_j is an idle MCS, and ϕ_j can send the charging responses to nearby pending EVCs when the charging requests are received by ϕ_j.

Particularly, a charging consensus between ϕ_j and v_i can be reached when the following conditions are satisfied: (*i*) ϕ_j receives a charging request from v_i; (*ii*) ϕ_j sends back a charging response to v_i; (*iii*) v_i selects ϕ_j (the charging position of ϕ_j produces the shortest detour distance for v_i among the idle MCSs which have sent the charging responses to v_i).

If ϕ_j does not reach any charging consensuses with pending EVCs at the t-th time slot, and then the scheduling of ϕ_j is conducted by SA-MADRL until the next time slot.

C. Charging Tasks

Assuming that an idle MCS ϕ_j reaches a charging consensus with a pending EVC v_i at the t-th time slot, the charging position $p_c(\phi_j)$ is selected from the charging parking lots and can produce the shortest detour distance for v_i:

$$\begin{aligned} &\arg \min_{p_c(\phi_j) \in \mathcal{P}} Dtr(v_i, \phi_j) \\ &s.t.\ e(v_i)^{(t)} \geq c \cdot d(v_i, p_c(\phi_j)), \end{aligned} \tag{1}$$

where $Dtr(v_i, \phi_j) = d(v_i, p_c(\phi_j)) + d(p_c(\phi_j), \ell_i) - d(v_i, \ell_i)$ denotes the detour distance of v_i charged at the position $p_c(\phi_j)$.

The charging delay of v_i is calculated by:

$$\xi(v_i) = \frac{Dtr(v_i, \phi_j)}{ms(v_i)} + T_W(v_i, \phi_j) + \frac{St(v_i) + c \cdot Dtr(v_i, \phi_j)}{\omega}, \tag{2}$$

where $\frac{St(v_i)+c \cdot Dtr(v_i,\phi_j)}{\omega}$ is the charging time for transferring electricity from ϕ_j to v_i, and ω denotes the charging speed. $T_W(v_i, \phi_j)$ denotes the possible delay of waiting for the arrival of ϕ_j:

$$T_W(v_i, \phi_j) = \max\left(0, \frac{d(\phi_j, p_c(\phi_j))}{ms(\phi_j)} - \frac{d(v_i, p_c(\phi_j))}{ms(v_i)}\right). \tag{3}$$

The charging revenue earned by ϕ_j is expressed as:

$$Rev(\phi_j) = \begin{array}{l} (\varrho_f - \varrho_0) \cdot [St(v_i) + c \cdot Dtr(v_i, \phi_j)] \\ -\varrho_0 \cdot d(\phi_j, p_c(\phi_j)) \end{array}, \tag{4}$$

where ϱ_f and ϱ_0 denote the unit price of electricity transferred from MCSs to EVCs and the unit price of electricity purchased from power grid, respectively.

The status transitions of MCSs and EVs are depicted by Fig. 4 involving the process from a quasi EVC detecting the low battery state to completing the charging task, and the process from an idle MCS finding EVCs to completing the charging task. Additionally, the arrow bars between the two status transition lines show the process of idle MCSs and pending EVCs reaching the charging consensuses.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410

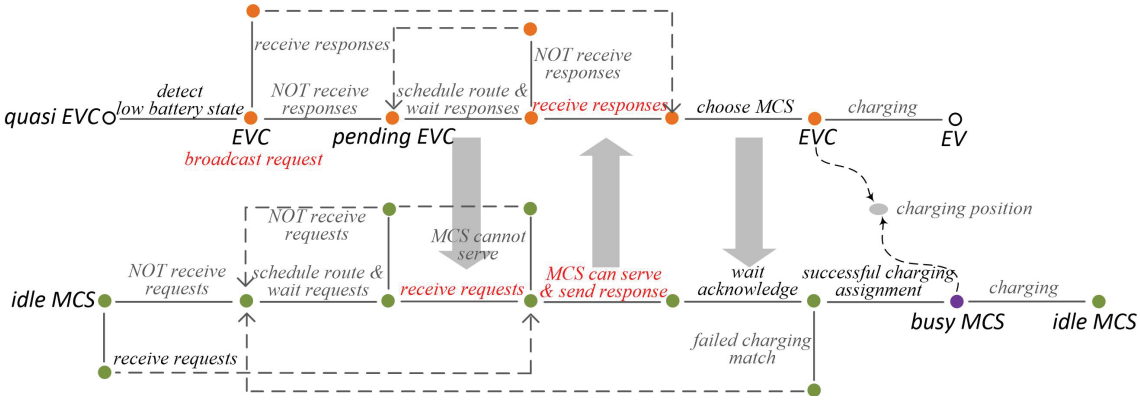


Fig. 4: Status transitions of MCSs and EVs.

D. Objective Functions

Our scheduling objectives are to increase the charging revenue of MCSs and enhance the proportion of successfully charged EVCs, and the scheduling objectives are formally presented as follows:

$$\begin{cases} \max & \sum_{\phi_j \in \mathcal{M}} \sum_{t=0}^{T} Rev(\phi_j), \\ \max & \frac{N_c}{N_e}, \end{cases} \tag{5}$$

where N_e and N_c denote the total number of EVCs and the number of successfully charged EVCs, respectively. T denotes the number of time slots in an episode. In the scheduling problem, the movements of MCSs and EVs are restricted by the road network, and MCSs must charge EVCs at charging parking lots, i.e., the charging positions must be selected from charging parking lots. Moreover, we assume that each MCS can charge a single EVC simultaneously.

IV. Multi-Agent Deep Reinforcement Learning Model

In our proposed SA-MADRL, MADRL is employed to address the scheduling problem of idle MCSs and pending EVCs, and the scheduling process is mapped into a sequential decision-making process, i.e., the scheduling routes (each scheduling route is constituted by the optimal scheduling points in multi-steps) are generated through a multi-step decision-making process. Due to the constraint imposed by the limited range of V2V communications, each agent is assumed to obtain the information in the local observation (within its communication range). Consequently, the scheduling problem can be formulated into a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [30].

A. Decentralized Partially Observable Markov Decision Process

Dec-POMDP is usually expressed as a septuple $< \mathcal{G}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathscr{P}, \mathscr{R}, \gamma >$, including the set of agents $\mathcal{G}$, set of states $\mathcal{S}$, set of joint observations $\mathcal{O}$, set of joint actions $\mathcal{A}$, state transition probability function $\mathscr{P}$, reward function $\mathscr{R}$, and discount factor γ. The detailed explanation of the components of Dec-POMDP is given as follows:

(i) Agents ($\mathcal{G}$): In the mobile charging scenario, the agents include all MCSs (idle MCSs and busy MCSs) and all EVCs (quasi EVCs and pending EVCs). The set of EVCs is expressed by $\{v_1, \cdots, v_n\}$, where n denotes the total number of quasi EVCs and pending EVCs.

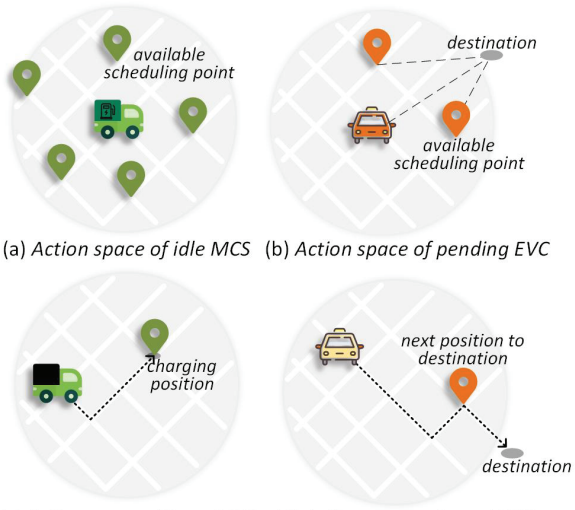
(ii) States ($\mathcal{S}$): The state at the t-th time slot is denoted by $s^{(t)}$ ($s^{(t)} \in \mathcal{S}$), which is comprised of the structure of the road network, and the real-time statuses of MCSs (positions and charging tasks) and EVs (positions and residual electricity).

(iii) Joint observations ($\mathcal{O}$): The local observation of the agent g_k at the t-th time slot is denoted by $o_k^{(t)}$ ($o_k^{(t)} \in \mathcal{O}$), which is a subset of $s^{(t)}$. Note that the range of local observation of each agent is equal to the communication range R_c. The joint observation at the t-th time slot is expressed as $\boldsymbol{o}^{(t)} = \left\{o_1^{(t)}, \cdots, o_k^{(t)}, \cdots, o_{n+M}^{(t)}\right\}$.

(iv) Joint actions ($\mathcal{A}$): At the t-th time slot, each agent (e.g., g_k) adopts the action $a_k^{(t)}$ based on the local observation $o_k^{(t)}$. A joint action $\boldsymbol{a}^{(t)} = \left\{a_1^{(t)}, \cdots, a_k^{(t)}, \cdots, a_{n+M}^{(t)}\right\}$ consists of the scheduling points in the local observations of all agents.

The action space (possible actions) of the agent g_k is denoted by $\mathcal{A}(g_k)^{(t)}$. As illustrated in Fig. 5, the action space of an idle MCS or a pending EVC consist of the available scheduling points in the local observation. Note that the available scheduling points of each pending EVC should satisfy that the travel distance from each available scheduling point to the destination of the pending EVC is shorter than the distance from the current position to the destination. This constraint ensures that pending EVCs can gradually approach their destinations while seeking the potential charging opportunities. For a quasi EVC, the action space only includes the next positions on the road network to its destination. Similarly, the action space of a busy MCS only includes the negotiated charging position.

(v) State transition probability function ($\mathscr{P}$): At the t-th time slot, each agent adopts an action under the state $s^{(t)}$, and after that the status of the agent could be altered. Additionally, some new charging requests could be launched, and some ongoing charging tasks could be



(a) *Action space of idle MCS* (b) *Action space of pending EVC*

(c) *Action space of busy MCS* (d) *Action space of quasi EVC*

Fig. 5: Action space.

completed. All these could lead to a transition from $s^{(t)}$ to a new state $s^{(t+1)}$, which obeys a state transition probability function $\mathscr{P}(s^{(t+1)}|s^{(t)}, \boldsymbol{a}^{(t)})$.

(*vi*) Reward function ($\mathscr{R}$): At the end of the t-th time slot, the agent g_k will obtain the reward $r_k^{(t)}$. The aim of each agent is to maximize the cumulative discounted future reward $\mathscr{R}(g_k) = \sum_{t=0}^{T} \gamma^t \cdot r_k^{(t)}$ (discounted return), where T represents the number of time slots in an episode.

Given a joint observation $\boldsymbol{o}^{(t)} = \left\{o_1^{(t)}, \cdots, o_{n+M}^{(t)}\right\}$, The value function of the agent g_k is expressed by the expectation of discounted return under a joint policy (strategy) $\pi = \{\pi_1, \cdots, \pi_{n+M}\}$:

$$\mathscr{F}_{k,\pi}(\boldsymbol{o}^{(t)}) = \sum_{\tau=t}^{T} \gamma^{\tau-t} \cdot \mathbb{E}_{\pi}\left[r_k^{(\tau)} | \ \boldsymbol{o}^{(t)}\right]. \tag{6}$$

Correspondingly, the action-value function is defined as $Q_{k,\pi}\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right)$, commonly referred to as the Q-function. $Q_{k,\pi}\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right)$ represents the expectation of discounted return under the strategy π when starting from the joint observation $\boldsymbol{o}^{(t)}$ and adopting the joint action $\boldsymbol{a}^{(t)}$:

$$Q_{k,\pi}\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right) = \sum_{\tau=t}^{T} \gamma^{\tau-t} \cdot \mathbb{E}_{\pi}\left[r_k^{(\tau)} | \ \boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right]. \tag{7}$$

By eliminating the influence of the strategy π, we can derive the optimal Q-function $Q_k^*\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right)$, which can obtain the maximum expectation of discounted return after adopting the joint action $\boldsymbol{a}^{(t)}$ under the observation $\boldsymbol{o}^{(t)}$:

$$Q_k^*\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right) = \max_{\pi} \ Q_{k,\pi}\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right). \tag{8}$$

The Bellman equation can be used to iterate the optimal Q-function:

$$Q_k^*\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}\right) = \mathbb{E}\left[r_k^{(t)} + \max_{\boldsymbol{a}^{(t+1)}} \ Q_k^*\left(\boldsymbol{o}^{(t+1)}, \boldsymbol{a}^{(t+1)}\right)\right]. \tag{9}$$

The calculation of the expectation in (9) is a challenging issue due to the model-free feature. Hence, we approximate $Q_k^*(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)})$ to a Q-network $Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right)$, where θ denotes the network parameters. Besides, to enhance the training stability of $Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right)$, the Q-target network [31] is adopted, and the parameters of the Q-target network (denoted by θ') are regularly copied from θ (every κ time slots).

B. Mean-Field MADRL

Considering that all agents (at the t-th time slot) make the action decisions based on $\boldsymbol{o}^{(t)}$ and $\boldsymbol{a}^{(t)}$, and the dimensions of $\boldsymbol{o}^{(t)}$ and $\boldsymbol{a}^{(t)}$ grow proportionally to the number of agents. Thus, it is extremely difficult to learn $Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right)$. To this end, each agent should communicate with nearby agents and acquire their statuses and action intentions.

Note that different nearby agents have different influences on the action decision of an agent. Therefore, the weighted mean-field MADRL is applied to approximate the local interactions between each agent and nearby ones. This method largely reduces the computational complexity, and can achieve the effect of global interactions as much as possible [32]. Consequently, the Q-network $Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right)$ can be factorized as:

$$Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right) = \sum_{g_{k'} \in \mathcal{N}(g_k)} w_{k,k'} \cdot Q_k\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}; \theta\right), \tag{10}$$

where $\mathcal{N}(g_k)$ denotes the set of nearby agents of g_k, and $w_{k,k'}$ signifies the influence of the agent $g_{k'}$ on the agent g_k.

The feature representation of the observation-action pair $\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}\right)$ indicates that the observation of the agent g_k is $o_k^{(t)}$, and g_k adopts the action $a_k^{(t)}$, while a nearby agent $g_{k'}$ adopts the action $a_{k'}^{(t)}$. The pairwise interaction Q-network $Q_k\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}; \theta\right)$ can be approximated by the mean field theory. The weighted average observation-action pair for g_k is denoted by $\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)$, which is transformed from $\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}\right)$ by a small fluctuation $\delta \cdot a_{k,k'}$:

$$\begin{aligned}\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right) &= \sum_{g_{k'} \in \mathcal{N}(g_k)} w_{k,k'} \cdot \left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}\right) \\ &= \sum_{g_{k'} \in \mathcal{N}(g_k)} w_{k,k'} \cdot \left[\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right) + \delta \cdot a_{k,k'}\right].\end{aligned} \tag{11}$$

Under the assumption of twice-differentiability, $Q_k\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}; \theta\right)$ can be expanded at $\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)$

using Taylor's theorem:

$$
\begin{aligned}
Q_k\left(\boldsymbol{o}^{(t)}, \boldsymbol{a}^{(t)}; \theta\right) &= \sum_{g_{k'} \in \mathcal{N}(g_k)} w_{k,k'} \cdot Q_k\left(o_k^{(t)}, a_k^{(t)}, a_{k'}^{(t)}; \theta\right) \\
&= \sum_{g_{k'} \in \mathcal{N}(g_k)} w_{k,k'} \cdot \left\{ \begin{array}{l} Q_k\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}; \theta\right) \\ +\delta \cdot a_{k,k'} \cdot \nabla_{\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}\right)} \\ \cdot Q_k\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}; \theta\right) \\ + \frac{\left(\delta \cdot a_{k,k'} \cdot \nabla_{\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}\right)}\right)^2}{2!} \\ \cdot Q_k\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}; \theta\right) \end{array} \right\} \\
&= Q_k\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}; \theta\right) + \sum_{g_{k'} \in \mathcal{N}(g_k)} \frac{w_{k,k'}}{2} \cdot Q_k\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}; \theta\right) \\
&\cdot \left(\delta \cdot a_{k,k'} \cdot \nabla_{\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}\right)}\right)^2 \approx Q_k\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}; \theta\right),
\end{aligned} \tag{12}
$$

where $Q_k(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}; \theta) \cdot \left(\delta \cdot a_{k,k'} \cdot \nabla_{\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}\right)}\right)^2$ is the Lagrange remainder with $\left(o_k^{(t)}, a_k^{(t)}, \widehat{a}_{k,k'}^{(t)}\right) = \left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right) + \lambda_{k,k'} \cdot \delta \cdot, a_{k,k'}$ ($\lambda_{k,k'} \in [0,1]$), acting as a small fluctuation near zero. Therefore, only the weighted average observation-action pair $(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)})$ needs to be calculated.

C. Training of SA-MADRL

Since the number of agents on the road network is typically large and could be changed frequently, we reduce the computational overhead by making the agents with the same type to share the same Q-network parameters. The training of SA-MADRL is illustrated in Fig. 6.

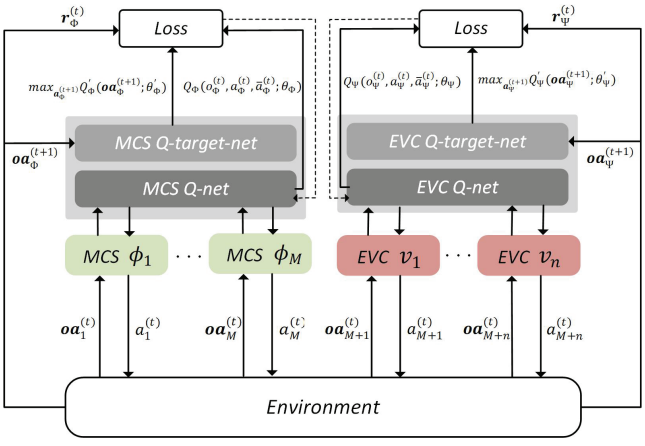


Fig. 6: Training of SA-MADRL.

Two types of agents, MCSs and EVCs train the Q-networks $Q_\Phi(;\theta_\Phi)$ and $Q_\Psi(;\theta_\Psi)$, respectively. Accordingly, there are two target-networks: $Q'_\Phi(;\theta'_\Phi)$ and $Q'_\Psi(;\theta'_\Psi)$. As shown in Fig. 6, at the t-th time slot, each agent (e.g., g_k) gets the set of observation-action pairs $\boldsymbol{oa}_k^{(t)} = \bigcup_{a_k^{(t)} \in \mathcal{A}(g_k)^{(t)}} \left\{\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)\right\}$, and $\boldsymbol{oa}_k^{(t)}$ denotes a set of observation-action pairs of g_k's action space (all available scheduling points).

In the following description, we assume that g_k is an MCS.

Firstly, the action $a_k^{(t)}$ is determined by the ε-greedy strategy:

$$
a_k^{(t)} = \left\{ \begin{array}{l} rand(scheduling\ points), \text{ with } \varepsilon, \\ \arg\max_{a_k^{(t)}} Q_\Phi(\boldsymbol{oa}_k^{(t)}; \theta_\Phi), \text{ with } 1-\varepsilon, \end{array} \right. \tag{13}
$$

where ε and $1-\varepsilon$ denote the probabilities of adopting the random action and $\arg\max_a\ Q_\Phi(a;\theta_\Phi)$, respectively. The random action is obtained by randomly selecting one position from the available scheduling points.

When the action $a_k^{(t)}$ is executed, the environment is updated, i.e., the set of future observation-action pairs $\boldsymbol{oa}_k^{(t+1)}$ and the reward $r_k^{(t)}$ will be returned at the $(t+1)$-th time slot. Then, the target Q-value $\zeta_k^{(t)}$ is expressed as:

$$
\zeta_k^{(t)} = r_k^{(t)} + \max_{a_k^{(t+1)}} Q'_\Phi(\boldsymbol{oa}_k^{(t+1)}; \theta'_\Phi), \tag{14}
$$

and the loss function is defined as:

$$
Loss_k^{(t)} = \left\{\zeta_k^{(t)} - Q_\Phi\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}; \theta_\Phi\right)\right\}^2. \tag{15}
$$

The Q-network can be updated by (15) every time slot using the temporal difference method to make the current Q-value close to the target Q-value.

D. Set of Observation-action Pairs

The set of observation-action pairs $\boldsymbol{oa}_k^{(t)}$ of the agent g_k at the t-th time slot consists of the feature vectors of available scheduling points in the local observation of g_k, and note that the feature vectors are influenced by the nearby agents of g_k. As aforementioned above, only idle MCSs and pending EVCs need to be scheduled. Hence, we specially discuss the set of observation-action pairs of two types of agents:

(1) With regard to a pending EVC v_i, within the communication range of v_i, suppose there are some available scheduling points constituting a set $\mathcal{A}(v_i)^{(t)}$, a busy MCS ϕ_j, and another pending EVC $v_{i'}$. Assume that v_i has spent the time $T_L(v_i)$ on looking for idle MCSs.

By the V2V communications, v_i obtains the information of the nearby busy MCS ϕ_j, including the current position $p(\phi_j)^{(t)}$, charging position $p_c(\phi_j)$ (taken as the action intention of ϕ_j), busy time $T_B(\phi_j^{(t)})$, and residual electricity $e(\phi_j)^{(\tilde{t})}$ after completing the ongoing charging task. Suppose ϕ_j currently undertakes the charging task of an EVC v_χ, and ϕ_j completes the charging for v_χ at the $\tilde{t}$-th time slot.

Thus, the residual electricity of ϕ_j after charging v_χ is calculated by:

$$
e(\phi_j)^{(\tilde{t})} = e(\phi_j)^{(t)} - c \cdot d(\phi_j, p_c(\phi_j)) - St(v_\chi). \tag{16}
$$

(16) implies that the busy MCS ϕ_j can charge v_i only if the residual electricity $e(\phi_j)^{(\tilde{t})}$ is larger than the electricity shortage $St(v_i)$. If v_i adopts an action a (scheduled to the

point a), the feature of nearby busy MCSs of v_i is calculated by:

$$fe_{_M}(v_i) = \sum_{\phi_j} \frac{e(\phi_j)^{(\tilde{t})}}{d(p_c(\phi_j), a)},\ s.t.\ e(\phi_j)^{(\tilde{t})} \geq St(v_i). \quad (17)$$

Likewise, suppose v_i receives the charging request from another pending EVC $v_{i'}$, and the charging request includes the next scheduling point $p_{next}(v_{i'})^{(t)}$ (taken as the action intention of $v_{i'}$) and the electricity shortage $St(v_{i'})$. The feature of nearby pending EVCs of v_i is written as:

$$fe_{_N}(v_i) = \sum_{v_{i'}} \frac{St(v_{i'})}{d(p_{next}(v_{i'})^{(t)}, a)}. \quad (18)$$

If v_i adopts the action a and is taken as the agent g_k, then the observation-action pair $\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)$ is obtained by:

$$\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right) = \left(d(v_i, a), fe_{_M}(v_i), fe_{_N}(v_i)\right). \quad (19)$$

and thus $\boldsymbol{oa}_k^{(t)} = \bigcup_{a_k^{(t)} \in \mathcal{A}(v_i)^{(t)}} \left\{\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)\right\}$.

(2) With regard to an idle MCS ϕ_j, the set of available scheduling points is denoted by $\mathcal{A}(\phi_j)^{(t)}$. Suppose there is another idle MCS $\phi_{j'}$ and a quasi EVC v_i within the communication range of ϕ_j. Note that $\phi_{j'}$ can provide its next scheduling point $p_{next}(\phi_{j'})^{(t)}$ (taken as the action intention) and the residual electricity $e_{next}(\phi_{j'})^{(t)}$ when reaching $p_{next}(\phi_{j'})^{(t)}$. Then, the feature of nearby idle MCSs of ϕ_j is expressed as:

$$fm_{_M}(\phi_j) = \sum_{\phi_{j'}} \frac{e_{next}(\phi_{j'})^{(t)}}{d(p_{next}(\phi_{j'})^{(t)}, a)}. \quad (20)$$

Besides, the feature of nearby quasi EVCs of ϕ_j is expressed as:

$$fm_{_N}(\phi_j) = \sum_{v_i} \frac{St(v_i)}{d(p_{next}(v_i)^{(t)}, a)}, s.t.\ e(\phi_j)^{(t)} \geq St(v_i), \quad (21)$$

where $p_{next}(v_i)^{(t)}$ denotes the next position on the road network to the destination of v_i. If ϕ_j adopts the action a and is taken as the agent g_k, then the observation-action pair $\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)$ is given by:

$$\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right) = \left(d(\phi_j, a), fm_{_M}(\phi_j), fm_{_N}(\phi_j)\right), \quad (22)$$

and thus $\boldsymbol{oa}_k^{(t)} = \bigcup_{a_k^{(t)} \in \mathcal{A}(\phi_j)^{(t)}} \left\{\left(o_k^{(t)}, a_k^{(t)}, \overline{a}_k^{(t)}\right)\right\}$.

E. Reward

After the agents determine and adopt the actions at the t-th time slot, the state is transformed into $s^{(t+1)}$. Then, the reward for each agent can be calculated under the state $s^{(t+1)}$. Since that only the scheduling of idle MCSs and pending EVCs is considered in this paper, the following two cases (Fig. 7) are discussed:

Case *A*. When the agent g_k is a pending EVC v_i, the objective is to approach the nearby busy MCSs that are about to complete their ongoing charging tasks and provide potential charging services for v_i in future, while the nearby pending EVCs could compete with v_i for the potential charging services. Hence, we use the nearby busy MCSs and pending EVCs of g_k to evaluate the decided action $a_k^{(t)}$:

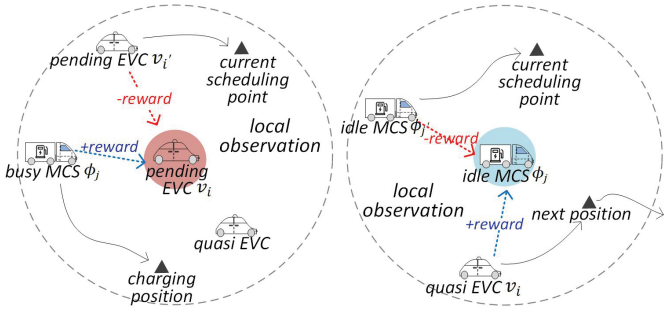


Fig. 7: Reward of a pending EVC and an idle MCS.

$$r_k^{(t)} = \sum_{\phi_j} \left[\frac{e(\phi_j)^{(\tilde{t})}}{d(g_k, \phi_j)} - \sigma \cdot \sum_{v_{i'}} \frac{St(v_{i'})}{d(v_{i'}, \phi_j)}\right], \quad s.t. \begin{cases} e(\phi_j)^{(\tilde{t})} \geq St(g_k), \\ e(\phi_j)^{(\tilde{t})} \geq St(v_{i'}), \end{cases} \quad (23)$$

where $\frac{e(\phi_j)^{(\tilde{t})}}{d(g_k,\phi_j)} - \sigma \cdot \sum_{v_{i'}} \frac{St(v_{i'})}{d(v_{i'},\phi_j)}$ denotes the reward of the nearby busy MCS ϕ_j, and σ is a preset weight. Specifically, $\frac{e(\phi_j)^{(\tilde{t})}}{d(g_k,\phi_j)}$ indicates that a larger reward is obtained when g_k is closer to the charging position of ϕ_j which has more residual electricity, or g_k is farther away from $v_{i'}$ which has smaller electricity shortage.

The constraints in (23) imply that: The nearby busy MCSs which could charge g_k in future and its competitors should be considered. By (23), a larger reward is obtained if a pending EVC is scheduled to a point around which there are more busy MCSs (more future charging opportunities) and fewer pending EVCs (smaller charging demand, and the charging competition can be relieved).

Case *B*. When the agent g_k is an idle MCS ϕ_j, the objective is to approach the quasi EVCs while considering the influences of other nearby idle MCSs. Then, $r_k^{(t)}$ is expressed as:

$$r_k^{(t)} = \sum_{v_i} \left[\frac{St(v_i)}{d(v_i, g_k)} - \rho \cdot \sum_{\phi_{j'}} \frac{e(\phi_{j'})^{(t+1)}}{d(v_i, \phi_{j'})}\right], \quad s.t. \begin{cases} e(g_k)^{(t+1)} \geq St(v_i), \\ e(\phi_{j'})^{(t+1)} \geq St(v_i), \end{cases} \quad (24)$$

where ρ is another preset weight. (24) indicates that a larger reward is obtained if an idle MCS is scheduled to a point around which there are more quasi EVCs which have larger electricity shortage and fewer busy MCSs which have less residual electricity.

V. Scheduling Approach Based on Multi-Agent Deep Reinforcement Learning

In order to increase the charging revenue of MCSs and enhance the proportion of successfully charged EVCs,

two types of Q-networks trained by the cloud server, are released to MCSs and EVs for their use, respectively. Thus, idle MCSs and pending EVCs can make the scheduling decisions independently. The details of SA-MADRL are introduced as follows:

A. Scheduling of Pending EVCs

Stage $A.1$: Every time slot each pending EVC (e.g., v_i) broadcasts a charging request within the communication range R_c (Fig. 8). The charging request $request\_msg$ is expressed as $\left(p_{next}(v_i)^{(t)}, St(v_i)\right)$, which includes the next scheduling point and electricity shortage of v_i.

Stage $A.2.1$. After receiving the $request\_msg$ of v_i, each nearby idle MCS (e.g., ϕ_j) will reply with an $approval\_msg$ to v_i (Fig. 8), and the $approval\_msg$ includes the current position and residual electricity of ϕ_j, i.e., $\left(p(\phi_j)^{(t)}, e(\phi_j)^{(t)}\right)$.

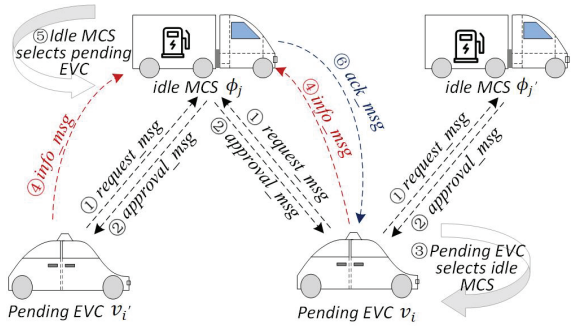


Fig. 8: Message exchanges between pending EVCs and idle MCSs.

Likewise, a busy MCS will reply a $status\_msg$ containing the charging position and residual electricity after completing the ongoing charging task, and note that v_i could also receive the $request\_msg$ from other pending EVCs.

Stage $A.2.2$. Suppose v_i receives several $approval\_msg$ from nearby idle MCSs (the set of these idle MCSs is denoted by $\mathcal{M}_I(v_i)$), and then v_i selects the optimal idle MCS and the charging position by:

$$\begin{aligned} &\arg_{\widetilde{\phi}\in\mathcal{M}_I(v_i),\widetilde{p}\in\mathcal{P}} \min\ Dtr(v_i,\widetilde{\phi}),\\ &s.t.\ \begin{cases} e(v_i)^{(t)} \geq c\cdot d(v_i,\widetilde{p}),\\ e(\widetilde{\phi})^{(t)} - c\cdot d(\widetilde{\phi},\widetilde{p}) \geq St(v_i) + c\cdot Dtr(v_i,\widetilde{\phi}), \end{cases} \end{aligned} \tag{25}$$

where $Dtr(v_i,\widetilde{\phi}) = d(v_i,\widetilde{p}) + d(\widetilde{p},\ell_i) - d(v_i,\ell_i)$. (25) implies that v_i will select the optimal idle MCS and the charging position to undertake the shortest detour distance, and the two constraints in (25) requires that: (i) The residual electricity of v_i can support the travel to the charging position; (ii) The residual electricity of idle MCS $\widetilde{\phi}$ can support the travel to the charging position and the electricity transferred to v_i.

After that, v_i sends an $info\_msg$ (Fig. 8) containing the charging position and detour distance to the selected idle MCS, and waits for the ack_msg (Fig. 8) from the selected idle MCS.

Stage $A.2.3$. If v_i does not receive any ack_msg from idle MCSs, indicating that v_i will remain a pending EVC within the t-th time slot. When v_i can arrive at the current scheduling point within the t-th time slot, a new scheduling point (an action) should be decided for v_i by the EVC Q-network. Specifically, the set of observation-action pairs input into the Q-network consists of the features of available scheduling points, which are calculated according to the information in the $status\_msg$ of busy MCSs and the $request\_msg$ of other pending EVCs. The scores of available scheduling points are obtained by the EVC Q-network. Then, v_i selects the point with the highest score as the new scheduling point, and moves towards this scheduling point.

B. Scheduling of Idle MCSs

Stage $B.1$: For each idle MCS (e.g., ϕ_j), if ϕ_j has received one or more $info\_msg$ from nearby pending EVCs (these EVCs constitute a set $\mathcal{N}_p(\phi_j)$), and then selects the optimal pending EVC which brings the largest charging revenue:

$$\arg_{\widetilde{v}\in\mathcal{N}_p(\phi_j)} \max\ Rev(\phi_j). \tag{26}$$

Then, ϕ_j sends an ack_msg to the selected pending EVC, and they move towards the charging position for the charging.

Stage $B.2$: For the idle MCS ϕ_j, if ϕ_j does not receive any $info\_msg$ from nearby pending EVCs, indicating that ϕ_j will remain an idle MCS within the t-th time slot, and a new scheduling point (an action) should be decided for ϕ_j by the MCS Q-network. Note that when ϕ_j reaches the current scheduling point, it should send an $inqury\_msg$ to the nearby idle MCSs to get their $status\_msg$ (including their current scheduling points and residual electricity). The set of observation-action pairs of ϕ_j can be formed according to the available scheduling points, $status\_msg$ of other idle MCSs, and $demand\_msg$ (similar to $request\_msg$) of quasi EVCs. Then, ϕ_j obtains the new scheduling point with the highest score, and ϕ_j moves towards the new scheduling point.

C. Charging Assignments for Quasi EVCs

Stage $C.1$: With regard to each quasi EVC (e.g., $v_{i'}$), $v_{i'}$ broadcasts a $demand\_msg$ (similar to $request\_msg$) within the communication range R_c. $v_{i'}$ could be charged by a nearby idle MCS before it turns into a pending EVC.

The sequential diagram example of message exchanges in SA-MADRL is illustrated in Fig. 9, where the quasi EVC v_1 broadcasts the charging demand (a $demand\_msg$), and two pending EVCs v_2 and v_3 broadcast the charging requests $request\_msg$. When v_2 receives two $approval\_msg$ from idle MCSs ϕ_1 and ϕ_2, v_2 chooses ϕ_2 and reaches a charging consensus with ϕ_2. Another idle MCS ϕ_1 (not reaching any charging consensuses) during the current time slot decides a new scheduling point by SA-MADRL. For the pending EVC v_3 which doesn't receive any $approval\_msg$ after broadcasting a $request\_msg$, it

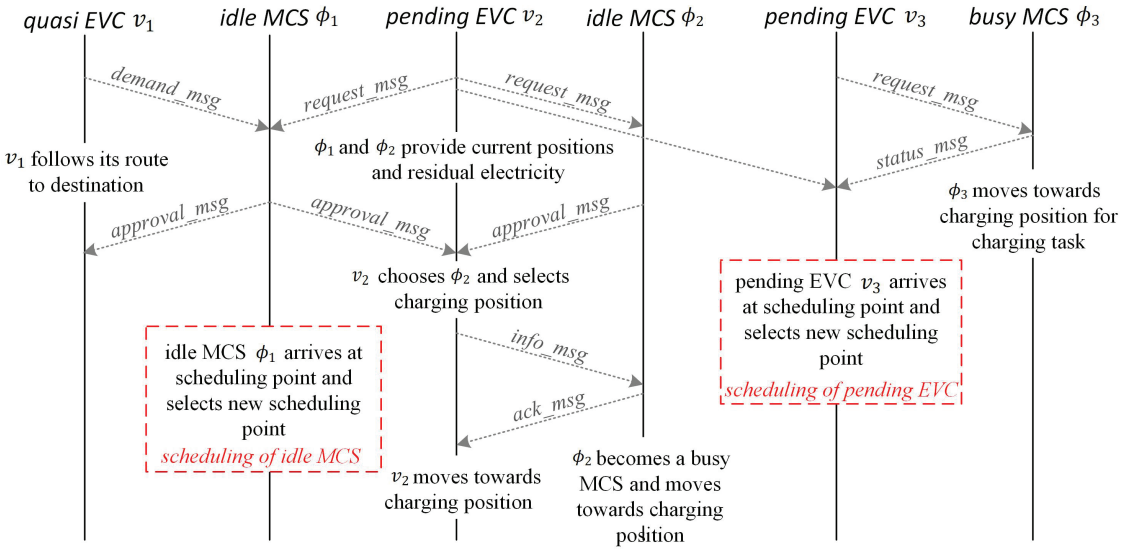


Fig. 9: A sequential diagram example of message exchanges in SA-MADRL.

employs SA-MADRL to decide the new scheduling point as well. In addition, the busy MCS ϕ_3 replies with a $status\_msg$ after receiving the $request\_msg$ from v_3.

VI. Performance Evaluations

In this section, we provide some performance evaluations on our proposed SA-MADRL. For the simulations, we obtain the map data from OpenStreetMap [34] using OSMnx [35] in the area with longitude interval [103.979,104.163] and latitude interval [30.597, 30.731] (part of Chengdu city, China). We filter out 338 scheduling points that are distributed evenly on the road network (obtained from the map data).

The initial battery capacity of each MCS is 100 kwh, and that of each EV obeys a normal distribution $N(\mu, \epsilon^2)$ [33], where μ denotes the average residual electricity of EVs, and ϵ denotes the deviation of residual electricity among EVs. The position of each EV (which has not made a charging request) is sampled every time slot from a real-world taxi dataset provided by Didi Corporation [36]. This dataset contains the GPS trajectories of more than 10,000 taxis during the period from October 1, 2018 to October 31, 2018 in Chengdu city. Each GPS trajectory is represented by a sequence of taxi ID, latitudes, longitudes, and timestamps. We assume that when the residual electricity of each MCS falls below the low battery state after completing the charging tasks or during the scheduling process, the MCS will be offline during the subsequent 5 time slots (the MCS is assumed to be charged by the grid or FCSs during the 5 time slots).

In SA-MADRL, two Q-networks (and two target Q-networks) are trained for idle MCSs and pending EVCs, respectively. These Q-networks have the same network structure, consisting of three fully connected layers with an activation function ReLU between them. The input layer receives the feature matrix $\boldsymbol{oa}_k^{(t)}$. The dimension of the hidden layer is expanded to 20, and the output dimension is set to 1. The parameters of the Q-target networks are regularly copied from the Q-networks every 10 time slots.

The scheduling points and trajectory data together form the simulated environment for the offline training of idle MCSs and pending EVCs. An example is illustrated in Fig. 10, where the scheduling of idle MCSs and pending EVCs decided by SA-MADRL are marked. Fig. 10 shows that the process of idle MCSs being attracted by some quasi EVCs and the process of pending EVCs approaching some busy MCSs.

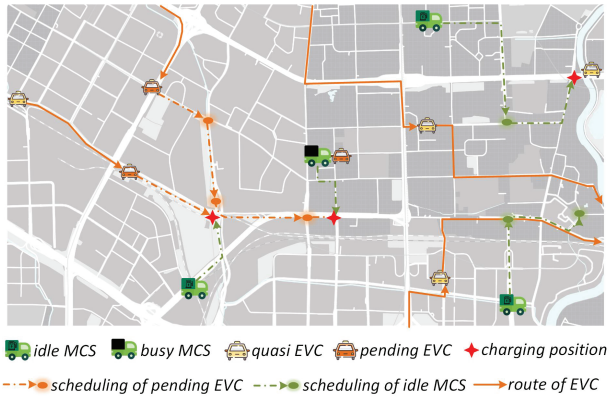


Fig. 10: An example of scheduling of idle MCSs and pending EVCs decided by SA-MADRL.

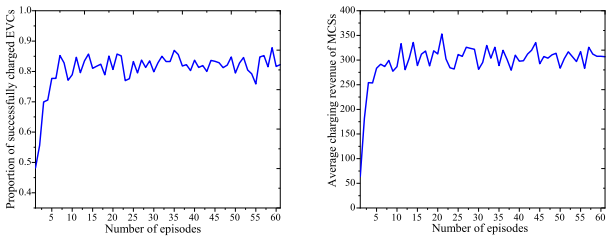
The main parameter settings are shown in TABLE III.

A. Convergence of SA-MADRL

The convergence plots of SA-MADRL are presented in Fig. 11, which reflects the convergence of the two objective functions. The x-axis denotes the number of episodes in training, and the y-axis denotes the proportion of successfully charged EVCs (Fig. 11(a)) and average charging revenue of MCSs (Fig. 11(b)), respectively.

TABLE III: Simulation parameters

Parameter	Description	Value
N	Number of EVs	500
M	Number of MCSs	20
$\wp$	Number of charging parking lots	338
t_s	Length of each time slot	5 min
R_c	Communication range of EVs and MCSs	2 km
T	Number of time slots in an episode	100
D	Maximum charging delay	10 min
ω	Charging speed	120 kwh/h
$ms(v_i)$, $ms(\phi_j)$	Travel speed of EVs and MCSs	39.6 km/h
μ	Average residual electricity of EVs	40 kwh
ϵ	Standard deviation of residual electricity	14 kwh
c	Electricity consumption for travelling through a unit distance	0.3 kwh/km
ϱ_0	Unit price of electricity purchased from power grid	0.5 /kwh
ϱ_f	Unit price of electricity transferred from MCSs to EVCs	1.6 /kwh
$\mathcal{L}$	Low battery state	8 kwh
γ	Discount factor	0.9
ε	Parameter in ε-greedy strategy	0.1
κ	Update interval of Q-target network parameters (the number of time slots)	10
σ	Preset weight	0.5
ρ	Preset weight	0.1



(a) Proportion of successfully charged EVCs vs. number of episodes

(b) Average charging revenue of MCSs vs. number of episodes

Fig. 11: Convergence of SA-MADRL.

During the first 7 episodes, the agents' scheduling policies are rapidly improved, leading to a notable increase in the proportion of successfully charged EVCs and the average charging revenue of MCSs (in an episode). In the subsequent episodes, the proportion of successfully charged EVCs stabilizes around 0.82, and the average charging revenue of MCSs stabilizes around 306. These results imply that our proposed SA-MADRL can converge quickly.

B. Proportion of Successfully Charged EVCs and Average Charging Revenue of MCSs

The proportion of successfully charged EVCs can reflect the charging experience of EV drivers. Fig. 12(a) illustrates the impacts of M and N on the proportion of successfully charged EVCs. The proportion of successfully charged EVCs is decreased with the increase of N because more EVCs could compete for the charging services of MCSs, and thus more EVCs could not be charged by MCSs. On the contrary, the proportion of successfully charged EVCs is increased with the increase of M, and the reason is that EVCs are easier to be charged when MCSs are deployed more densely. Fig. 12(b) illustrates the impact of D on the proportion of successfully charged EVCs. D denotes the maximum charging delay allowing each pending EVC to be charged by MCSs after it makes a charging request. The proportion of successfully charged EVCs is gradually increased with the increase of D, because EVCs are allowed to wait for MCSs for a longer period.

Similar to Fig. 12(a), Fig. 12(c) delineates the impacts of M and N on the average charging revenue of MCSs. As M increases, there is a discernible decrease in the average charging revenue, whereas an increment in N corresponds to an elevation in the average charging revenue. In Fig. 12(d), the curve with a larger μ is lower than that with a smaller one. This phenomenon is attributed to the fact that EVCs with more residual electricity require less electricity, consequently leading to a reduction in the charging revenue of MCSs. Likewise, a smaller ϵ implies a greater proportion of EVs possessing sufficient residual electricity, thereby diminishing the charging revenue of MCSs as well.

C. Impacts of Learning Rate and ε

In Fig. 13, we first vary the learning rate $(0.05, 0.01, 0.005, 0.001)$ to observe the impacts on the performance of SA-MADRL, in terms of proportion of successfully charged EVCs and average charging revenue of MCSs.

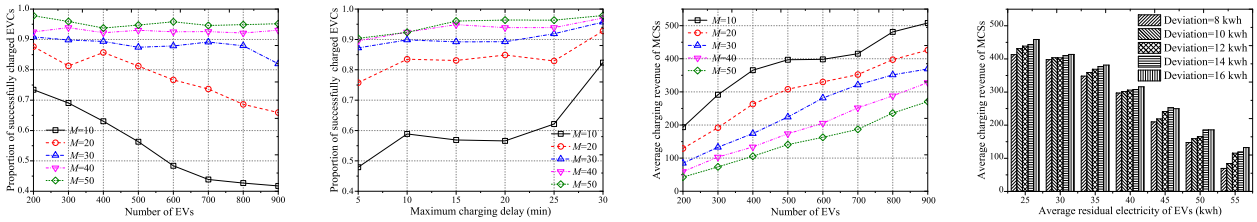
When the learning rate is too large (e.g. 0.05), SA-MADRL is difficult to converge. Conversely, with a learning rate of 0.001, the convergence speed is slower than that of 0.005 or 0.01, but the curve with 0.001 is more stable after the convergence. When the learning rate is set to 0.005 or 0.01, SA-MADRL rapidly converges after the 6-th episode. With the learning rate of 0.005, the proportion of successfully charged EVCs (the average value after the 6-th episode) reaches 0.84, and the average charging revenue of MCSs (the average value after the 6-th episode) reaches 325.

ε denotes the probability of the agent selecting a random action when making decisions. The impacts of ε are also observed. When ε is set to 0.1, the fastest convergence can be achieved, and the best performance is obtained. When ε is set to 0.3 the curve exhibits significant fluctuations, and when ε is set to 0.01 the performance is poor after the convergence. These observations are attributed to the following facts: Too large ε increases the randomness in the agent's decision-making, making SA-MADRL converge slowly and the performance be unstable; Too small ε reduces the exploration opportunities of agents, making SA-MADRL hard to yield the optimal outcomes.

D. Average Charging Delay and Average Detour Distance of EVCs

In Fig. 14(a) and Fig. 14(c), we observe an increase in the average charging delay and average detour distance as

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410



(a) Proportion of successfully charged EVCs vs. N and M (b) Proportion of successfully charged EVCs vs. D and M (c) Average charging revenue of MCSs vs. N and M (d) Average charging revenue of MCSs vs. μ and ϵ

Fig. 12: Proportion of successfully charged EVCs and average charging revenue of MCSs.

N grows, due to the competition among EVCs. With more EVCs contending for the charging opportunities, more time and detour are consumed to find the idle MCSs. With an increase in the number of MCSs, both the average charging delay and average detour distance of EVCs are decreased, and this is because more MCSs provide more charging opportunities, EVCs are easier to be charged by MCSs.

Fig. 14(b) and Fig. 14(d) illustrate the impacts of D and M on the average charging delay and average detour distance of EVCs. As depicted in Fig. 14(b), a larger D allows EVCs to have more time to seek out the charging opportunities provided by MCSs, thus more EVCs can be charged (Fig. 12(b)), consequently resulting in larger charging delay and longer detour distance.

E. Average Cruising Time and Average Charging Cost of MCSs

The cruising time of MCSs consists of the time the MCSs in the idle state and the time for MCSs moving to the charging positions after acknowledging the charging tasks. Fig. 15(a) exhibits an increase in the average cruising time with the rise in N, and a decrease in the average cruising time with the rise in M, because more EVCs and fewer MCSs make each MCS should undertake more charging tasks, thus shortening the cruising time.

Fig. 15(b) illustrates the impacts of μ and ϵ. A larger μ results in the reduction of electricity shortage of EVCs, thereby shortening the charging time of MCSs and prolonging the cruising time of MCSs. On the contrary, a larger ϵ implies that more EVs turn into EVCs, necessitating MCSs to provide more charging services and reduce the cruising time.

The charging cost of MCSs includes the cost of electricity consumed on the cruises and the cost of electricity transferred to EVCs. In Fig. 15(c), as N increases, the number of EVCs increases, causing MCSs to produce higher charging cost for charging EVCs. Moreover, shorter idle time diminishes the cruising cost, and deploying more MCSs (larger M) gives rise to a decrease in the average charging cost of MCSs. Similar to Fig. 15(b), a larger μ leads to lower charging cost, whereas a larger ϵ leads to higher charging cost (Fig. 15(d)).

F. Comparisons among Different Strategies

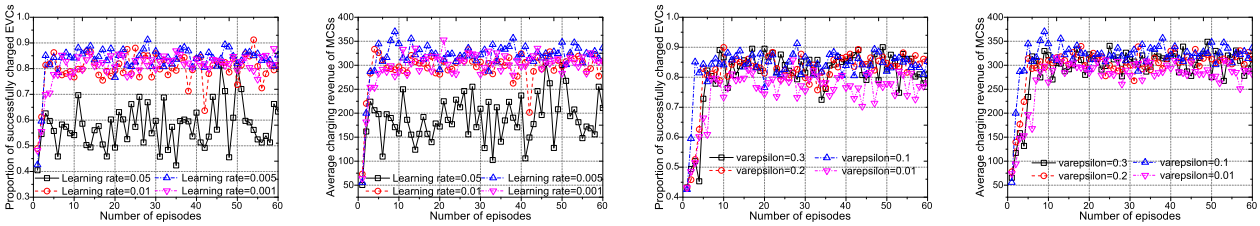
To further verify the merits of SA-MADRL, we compare SA-MADRL with some related strategies, such as Random Walk Strategy (RWS, idle MCSs move randomly, and pending EVCs move along their routes to destinations), SA-MADRL-E (idle MCSs move randomly, and pending EVCs are scheduled by SA-MADRL), SA-MADRL-M (idle MCSs are scheduled by SA-MADRL, and pending EVCs move along their routes to destinations), RLA [17], and O2O Mobile Charging System (O2OMCS) [22].

In these strategies, RLA is a scheduling strategy specifically designed for the peak charging periods. However, the training of our proposed SA-MADRL should be executed over 100 consecutive time slots, making it impossible to compare the training time with RLA. Additionally, O2OMCS and RWS are online decision-making strategies which do not require the training process. Therefore, we only compare the training time among SA-MADRL, SA-MADRL-M, and SA-MADRL-E, and the results are provided in TABLE IV. As shown in TABLE IV, SA-MADRL converges in the 6-th episode, while both SA-MADRL-M and SA-MADRL-E converge in the 4-th episode. It is evident that the strategies (SA-MADRL-M and SA-MADRL-E) only focusing on scheduling either MCSs or EVCs can converge more rapidly. However, the performance of SA-MADRL-M and SA-MADRL-E is worse than that of SA-MADRL (Fig. 16). Note that most of the training time is spent on updating the environment, and the training time consumed in each episode is almost the same, although the number of training iterations in each episode could be quite different.

TABLE IV: Training complexity

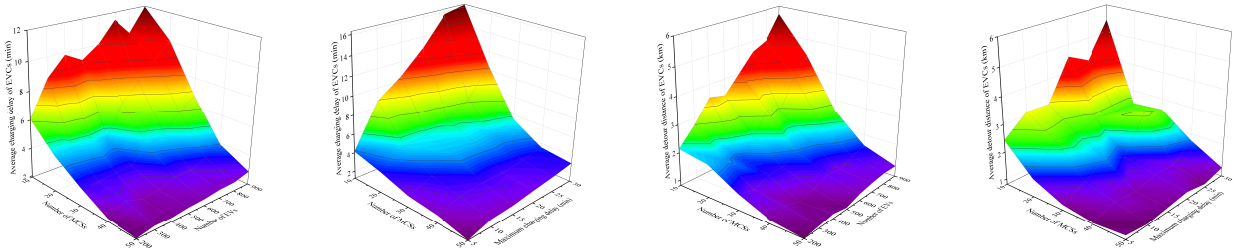
Training complexity	SA-MADRL	SA-MADRL-M	SA-MADRL-E
Training time (s)	215.23	139.95	141.67
Number of training iterations of MCS Q-network	281	201	0
Number of training iterations of EVC Q-network	113	0	98
Number of episodes for convergence	6	4	4

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3373410



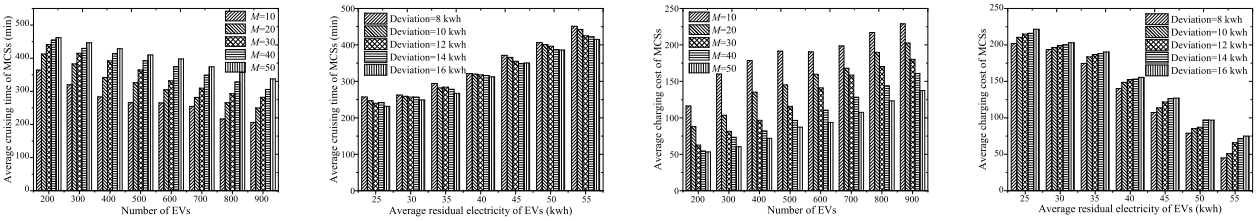
(a) Proportion of successfully charged EVCs vs. learning rate (b) Average charging revenue of MCSs vs. learning rate (c) Proportion of successfully charged EVCs vs. ε (d) Average charging revenue of MCSs vs. ε

Fig. 13: Impacts of learning rate and ε.



(a) Average charging delay of EVCs vs. N and M (b) Average charging delay of EVCs vs. D and M (c) Average detour distance of EVCs vs. N and M (d) Average detour distance of EVCs vs. D and M

Fig. 14: Average charging delay and average detour distance of EVCs.



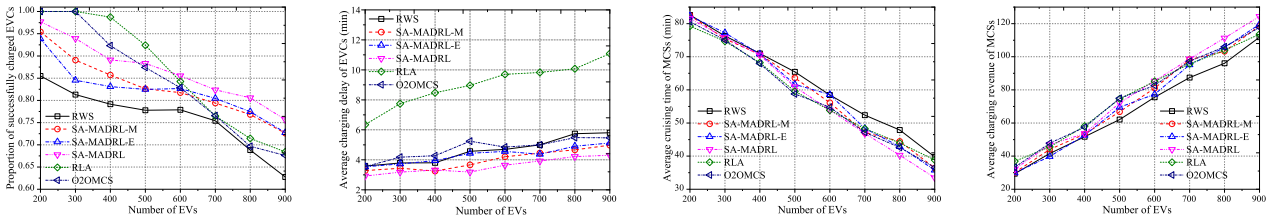
(a) Average cruising time of MCSs vs. N and M (b) Average cruising time of MCSs vs. μ and ϵ (c) Average charging cost of MCSs vs. N and M (d) Average charging cost of MCSs vs. μ and ϵ

Fig. 15: Average cruising time and average charging cost of MCSs.

These strategies are compared in terms of the proportion of successfully charged EVCs, average charging delay of EVCs, average cruising time of MCSs, and average charging revenue of MCSs. Since RLA is specially designed for the peak charging periods, we only select 20 time slots with high charging demand for the following simulations. The simulation results are given in Fig. 16.

The simulation results obtained by SA-MADRL-M and SA-MADRL-E demonstrate that scheduling either idle MCSs or pending EVCs can improve the charging performance, especially compared with RWS. The better outcomes are obtained when idle MCSs or pending EVCs are scheduled by SA-MADRL. The centralized strategies RLA and O2OMCS have better performance in terms of the proportion of successfully charged EVCs, average cruising time of MCSs, and average charging revenue of MCSs when $N < 500$. This is attributed to the fact that a centralized method typically facilitates the optimal scheduling of idle MCSs. However, with the increase in the number of EVCs, our proposed SA-MADRL properly schedules both idle MCSs and pending EVCs, and the charging requests launched by EVCs can be responded by idle MCSs more promptly. Moreover, note that a distributed method has lower communication complexity and computational complexity in real-world mobile charging scenarios.

RWS yields the smallest proportion of successfully charged EVCs, the longest average cruising time of MCSs, and the smallest average charging revenue of MCSs. These phenomena indicate that random movements for idle MCSs are quite not reasonable. RLA obtains the longest average charging delay of EVCs, since RLA pays more attention to the balance between charging supply and charging demand among different regions, and idle MCSs could travel for longer distance and spend more time on scheduling. Besides, RLA makes pending EVCs to actively move towards idle MCSs, implying that idle MCSs could wait for the arrivals of EVCs at the scheduling points, by implementing a charging queuing strategy. The charging delay of EVCs



(a) Proportion of successfully charged EVCs (b) Average charging delay of EVCs (c) Average cruising time of MCSs (d) Average charging revenue of MCSs

Fig. 16: Comparisons among different strategies (during 20 time slots with high charging demand).

includes the time of arriving at the scheduling points of the idle MCSs and the waiting time in the charging queues (the charging queues are typically long), thus resulting in the longest charging delay.

For the distributed scheduling strategies (SA-MADRL, SA-MADRL-M, and SA-MADRL-E), the decision-making time denotes the average duration required for each MCS or EV to reach a scheduling decision. The decision-making time of the centralized scheduling strategies RLA and O2OMCS includes the time of deciding the scheduling points of all MCSs and the time consumed on the order assignments. Notably, RLA selects the optimal regions for MCSs using a trained two-dimensional Q-table, thus facilitating a quicker decision-making response. O2OMCS needs to measure the operation utility at the available scheduling points of each MCS. In addition, the decision-making time of the centralized scheduling strategies obviously increases as the number of MCSs increases, and this phenomenon is different from that of the distributed ones, as illustrated in Fig. 17.

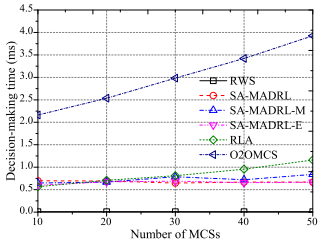


Fig. 17: Decision-making time.

The above results show that when the density of EVs on the road network is large (the number of EVs is large), our proposed SA-MADRL has better performance in terms of the proportion of charged EVCs, average charging delay of EVCs, average cruising time of MCSs, and average charging revenue of MCSs.

VII. Conclusion

We have studied the scheduling problem of idle MCSs and pending EVCs, and the Scheduling Approach based on Multi-Agent Deep Reinforcement Learning (SA-MADRL) has been introduced. In SA-MADRL, each idle MCS or each pending EVC obtains the surrounding situation regarding the charging supply and charging demand through V2V communications, and then uses a Q-network trained by multi-agent deep reinforcement learning method to make the scheduling decision independently. Therefore, SA-MADRL can enhance the proportion of successfully charged EVCs and increase the charging revenue of MCSs.

There are also some practical issues that need to be considered in future when applying our proposed SA-MADRL: (i) When scheduling idle MCSs and pending EVCs, FCSs are not considered in SA-MADRL. Both FCSs and MCSs are important components of IoEV. Specially, FCSs can be incorporated into the local observations of idle MCSs and pending EVCs. (ii) In real-world mobile charging scenarios, a single MCS can charge multiple EVCs simultaneously, which implies that busy MCSs possess the ability to charge more EVCs, i.e., a busy MCS having not started the charging process can negotiate with the served EVC to alter the charging position when receiving more charging requests from other pending EVCs. In addition, a busy MCS having started the charging process can provide free charging ports for other EVCs.

Acknowledgments

This research is supported by National Natural Science Foundation of China under Grant Nos. 62272237, 62372249, 61872191.

References

[1] I. Husain, B. Ozpineci, M. S. Islam, *et al*, "Electric drive technology trends, challenges, and opportunities for future electric vehicles," *Proceedings of the IEEE* , vol. 109, no. 6, pp. 1039–1059, 2021.

[2] H. S. Das, M. M. Rahman, S. Li, *et al*, "Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review," *Renewable and Sustainable Energy Reviews*, vol. 120, 2020.

[3] L. Pan, E. Yao, Y. Yang, *et al*, "A location model for electric vehicle (EV) public charging stations based on drivers existing activities," *Sustainable Cities and Society*, vol. 59, 2020.

[4] S. Afshar, P. Macedo, F. Mohamed, *et al*, "Mobile charging stations for electric vehicles–A review," *Renewable and Sustainable Energy Reviews*, vol. 152, 2021.

[5] X. Tang, S. Bi, and Y. Zhang, "Distributed routing and charging scheduling optimization for internet of electric vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 136–148, 2019.

[6] A. Pal, A. Bhattacharya, and A. K. Chakraborty, "Allocation of electric vehicle charging station considering uncertainties," *Sustainable Energy, Grids and Networks*, vol. 25, 2021.

[7] R. Saadati, J. Saebi, and M. Jafari-Nokandi, "Effect of uncertainties on siting and sizing of charging stations and renewable energy resources: A modified capacitated flow-refueling location model," *Sustainable Energy, Grids and Networks*, vol. 31, 2022.

[8] R. Saadati, M. Jafari-Nokandi, and J. Saebi, "Allocation of RESs and PEV fast-charging station on coupled transportation and distribution networks," *Sustainable Cities and Society*, vol. 65, 2021.

[9] R. Das, Y. Wang, K. Busawon, *et al*, "Real-time multi-objective optimisation for electric vehicle charging management," *Journal of Cleaner Production*, vol. 292, 2021.

[10] Q Chen, F. Wang, B. M. Hodge, *et al*, "Dynamic price vector formation model-based automatic demand response strategy for PV-assisted EV charging stations," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2903–2915, 2017.

[11] S. Limmer and T. Rodemann, "Peak load reduction through dynamic pricing for electric vehicle charging," *International Journal of Electrical Power & Energy Systems*, vol. 113, 2019.

[12] G. S. Aujla, N. Kumar, M. Singh, *et al*, "Energy trading with dynamic pricing for electric vehicles in a smart city environment," *Journal of Parallel and Distributed Computing*, vol. 127, 2019.

[13] C. Fang, H. Lu, Y. Hong, *et al*, "Dynamic pricing for electric vehicle extreme fast charging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 531–541, 2021.

[14] V. Moghaddam, A. Yazdani, H. Wang, *et al*, "An online reinforcement learning approach for dynamic pricing of electric vehicle charging stations," *IEEE Access*, vol. 8, pp. 130305–130313, 2020.

[15] T. Qian, C. Shao, X. Li, *et al*, "Multi-agent deep reinforcement learning method for EV charging station game," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1682–1694, 2022.

[16] C. D. Korkas, S. Baldi, S. Yuan, *et al*, "An adaptive learning-based approach for nearly optimal dynamic charging of electric vehicle fleets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2066–2075, 2018.

[17] J. Ni, R. Liang, and H. Wu, "Collaborative mobile charging vehicles placement: A reinforcement learning approach," *IEEE 23rd Int Conf on High Performance Computing & Communications*, Haikou, China, 2022.

[18] I. El-fedany, D. Kiouach, and R. Alaoui, "A smart coordination system integrates MCS to minimize EV trip duration and manage the EV charging, mainly at peak times," *International Journal of Intelligent Transportation Systems Research*, vol. 19, pp. 496–509, 2021.

[19] H. Ko, T. Kim, D. Jung, *et al*, "Software-defined electric vehicle (EV)-to-EV charging framework with mobile aggregator," *IEEE Systems Journal*, vol. 17, no. 2, pp. 2815–2823, 2023.

[20] L. Liu, Z. Xi, K. Zhu, *et al*, "Mobile charging station placements in Internet of electric vehicles: A federated learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24561–24577, 2022.

[21] H. Wang, R. Wang, H. Xu, *et al*, "Multi-objective mobile charging scheduling on the Internet of Electric Vehicles: a DRL approach," *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, 2022.

[22] P. Tang, F. He, X. Lin, *et al*, "Online-to-offline mobile charging system for electric vehicles: Strategic planning and online operation," *Transportation Research Part D: Transport and Environment*, vol. 87, 2020.

[23] J. Wang, Q. Guo, H. Sun, *et al*, "Collaborative optimization of logistics and electricity for the mobile charging service system," *Applied Energy*, vol. 336, 2023.

[24] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[25] Y. Zhou, S. Liu, Y. Qing, *et al*, "Is centralized training with decentralized execution framework centralized enough for MARL?," *arXiv:2305.17352*, 2023.

[26] R. Lowe, Y. Wu, A. Tamar, *et al*, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Conference and Workshop on Neural Information Processing Systems (NIPS)*, San Diego, USA, 2017.

[27] J. Foerster, G. Farquhar, T. Afouras, *et al*, "Counterfactual multi-agent policy gradients," *AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[28] T. Rashid, M. Samvelyan, C. S. D. Witt, *et al*, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7234–7284, 2020.

[29] Y. Yang, R. Luo, M. Li, *et al*, "Mean field multi-agent reinforcement learning," *35th International Conference on Machine Learning (PMLR)*, vol. 80, pp. 5571–5580, 2018.

[30] C. Amato, G. Chowdhary, A. Geramifard, *et al*, "Decentralized control of partially observable Markov decision processes," *52nd IEEE Conference on Decision and Control*, Firenze, Italy, 2013.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, *et al*, "Playing atari with deep reinforcement learning," *arXiv:1312.5602v1*, 2013.

[32] L. E. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, no. 3, pp. 387–424, 1993.

[33] R. R. Richardson, M. A. Osborne, and D. A. Howey, "Gaussian process regression for forecasting battery state of health," *Journal of Power Sources*, vol. 357, pp. 209–219, 2017.

[34] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[35] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

[36] Didi Corporation, "GAIA open dataset," *https://outreach.didichuxing.com/app-vue/dataList*, 2020.

AUTHOR BIOGRAPHY

Linfeng Liu received the B. S. and Ph. D. degrees in computer science from the Southeast University, Nanjing, China, in 2003 and 2008, respectively. At present, he is a Professor in the School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, China. His main research interests include the areas of vehicular ad hoc networks, wireless sensor networks and multi-hop mobile wireless networks. He has published more than 80 peer-reviewed papers in some technical journals or conference proceedings, such as IEEE TMC, IEEE TPDS, IEEE TIFS, IEEE TITS, IEEE TVT, IEEE TSC, ACM TAAS, ACM TOIT, Computer Networks, Elsevier JPDC. He has served as the TPC member of Globecom, ICONIP, VTC, WCSP.

Zhuo Huang received the B. S. degree in computer science from the Nanjing University of Posts and Telecommunications in 2022. At present, she is a master student of Nanjing University of Posts and Telecommunications. Her current research interest includes the areas of Internet of electric vehicles and vehicular ad-hoc networks.

Jia Xu received the Ph. D. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in Jiangsu Key Laboratory of Big Data Security and Intelligent Processing at Nanjing University of Posts and Telecommunications. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE, and has served as the Publicity Co-Chair of SciSec 2021.