Applied Energy 328 (2022) 120111

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

# Multi-agent deep reinforcement learning approach for EV charging scheduling in a smart grid

Keonwoo Park [a], Ilkyeong Moon [a,b,*]

[a] Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea
[b] Institute for Industrial Systems Innovation, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

ARTICLE INFO

*Keywords:*
Electric vehicles
Smart grid
Scheduling
Multi-agent deep reinforcement learning

ABSTRACT

As the competitive advantages of electric vehicles, both in terms of operating costs and eco-friendly characteristics have gained attention, the demand for electric vehicles has increased, and studies for efficiently charging electric vehicles are being actively conducted. Previous studies have mainly focused on scheduling one electric vehicle visiting a charging station or scheduling multiple electric vehicles in a centralized execution method. However, a decentralized execution method that can schedule multiple vehicles according to their status is more suitable in a realistic smart grid charging environment that requires quick decisions. Therefore, we propose a multi-agent deep reinforcement learning approach with a centralized training and decentralized execution method that can derive charging scheduling for each electric vehicle. Computational experiments show that the proposed approach shows desirable performance in minimizing the operating cost of electric vehicles.

## 1. Introduction

Owing to the eco-friendly and economic advantages of electric vehicles (EVs) compared to conventional internal combustion engine vehicles, the demand for EVs is on the rise. Therefore, the increased number of EVs consumes a sufficiently large amount of electricity within the grid. A smart grid system has emerged to solve the power supply problem for electric devices that use a large amount of electricity, including EVs. A smart grid is a two-way communication power supply system between the power company and users to efficiently control the usage of available electricity [1]. Various power systems (e.g., heating, ventilation, and air conditioning (HVAC) systems, home energy management systems (HEMS), and renewable energy systems) that consume a significant amount of energy are operated within the smart grid. The smart grid solves the power supply problems based on advanced communication equipment [2]. Using energy and its data from various systems generated in the smart grid can make the grid operator efficiently distribute electricity to consumers. Hence, with the growing interest in EVs and the development of smart grid technology, the field that considers charging scheduling to minimize the operating cost of EVs within the smart grid is receiving significant attention [3].

As the smart grid shows effective performance, EV charging stations in the smart grid, including solar power generation systems (PV) and energy storage systems (ESS), have recently appeared. Along with the increasing number of EV charging stations in the smart grid, research to efficiently operate the charging stations has also increased [4]. In order to efficiently operate EV charging stations, including complex systems, it is essential to respond appropriately to various variables occurring in the systems. Therefore, many studies have tried to minimize charging and operating costs by using optimization methods. However, optimizing the charging scheduling of EVs in the smart grid that communicates with various systems is not an easy problem. Considering uncertain situations within the smart grid (e.g., amount of PV generation, real-time electricity prices, and the arrival and departure times of EVs) is far from reality when dealing with the existing optimization method. However, using the data generated from the above situations can more practically approach the situation that occurs in a more realistic environment [5].

Reinforcement learning is a machine learning technique suitable for making sequential decisions in a complex environment in which much data is generated, such as in the smart grid. In reinforcement learning, the environment and the agent repeatedly interact by exchanging state, action, and reward. Unlike other algorithms operating in an environment where the model is completely known, reinforcement learning algorithms can learn in an environment without a model [6]. In particular, Sutton and Barto [6] showed that reinforcement learning derives an action that can maximize the reward in a given state and find the optimal decision, even in a situation where information is insufficient or uncertain. In addition, by overcoming the curse of dimensionality

[*] Corresponding author at: Department of Industrial Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea.
*E-mail addresses:* pgw9501@snu.ac.kr (K. Park), ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.apenergy.2022.120111
Received 13 July 2022; Received in revised form 31 August 2022; Accepted 5 October 2022
Available online 26 October 2022

**Table 1**
Commonly used abbreviations.

| Abbreviation | Description |
|---|---|
| EV | Electric vehicle |
| HVAC | Heating, ventilation and air conditioning |
| HEMS | Home energy management system |
| PV | Photovoltaic, solar power system |
| ESS | Energy storage system |
| SOC | State of charge |
| MADRL | Multi-agent deep reinforcement learning |
| COMA | Counter-factual multi-agent policy gradient |
| MADDPG | Multi-agent deep deterministic policy gradient |
| LSTM | Long short-term memory |
| CTDE | Centralized training and decentralized execution |

problem with a deep reinforcement learning technique grafted with a deep neural network, problems that were too large to handle can be solved efficiently [7]. With the above strengths, reinforcement learning can be considered suitable for making optimal decisions in the smart grid environment containing numerous and uncertain systems.

Therefore, we propose a MADRL (multi-agent deep reinforcement learning) method to efficiently charge multiple EVs that visit an EV charging station. Each as an agent, EVs participate in learning with their states. Based on the learning, each EV selects the charging or discharging action to be taken in the current state. We used the COMA (counter-factual multi-agent) policy gradient algorithm and MADDPG (multi-agent deep deterministic policy gradient) algorithm of the MADRL method. To be best of our knowledge, there has been no prior study of the charging scheduling method of multiple EVs in a smart grid. The contributions of this paper are threefold.

1. A multi-agent approach based on deep reinforcement learning is proposed to address the charging scheduling problem of multiple EVs. A multi-agent approach using centralized training and decentralized execution method gives faster and better solutions than existing methods.
2. In order to solve the situation in which multiple EVs visit a charging station in the smart grid, an EV charging station environment consisting of various power systems is presented. To model and solve the problem, we used the Markov decision process to formulate the problem as a sequential decision-making problem. Through this process, multiple EVs can get optimal energy at every time step according to their state respectively.
3. The proposed method can be practically applicable in that the method satisfies both EV users and an EV charging station operator. EV users can acquire the battery charge they want within a specified time. From the perspective of an EV charging station operator, the operator can minimize the operating cost while satisfying users.

The remainder of this study is organized as follows. In Section 2, literature review related to this paper is introduced. Section 3 provides explanations and mathematical formulations of the problem. Section 4 introduces the deep reinforcement learning algorithm and describes the specific process of how the MADRL algorithm is applied to the experiment. The computational experiments and their interpretation are presented in Section 5. Finally, the last section concludes this paper. A list of commonly used abbreviations is given in Table 1.

## 2. Literature review

Numerous charging scheduling approaches have been proposed to the electric power market in recent years. Developed from the one-way power supply system of the past, in which power grids supplied electricity to users, research on a two-way electricity exchange system that users (e.g., EV users) can supply with power in reverse is also being actively conducted [8]. In particular, discharging electricity from the EV to the power grid not only allows EV users to sell electricity at a high price but also can lower peak demand across the power grid. In addition, considering various power generation systems implemented in the smart grid is also a main concern in ongoing research [9].

Due to the rapid development of the electricity market, many companies in the electricity industry are focusing on and researching fields for practical implementation in the real world. Kempton et al. [10] and Mersky and Samaras [11] tested the impact of electricity costs on a two-way electricity exchange system with utility company PJM. Also, companies that make up the mainstream of electric vehicles (Tesla, Hyundai, etc.) are contemplating the optimal operation strategy for electric vehicles in the power grid [12,13].

Most of the previous studies tried to solve the charging scheduling problem by using an optimization method based on a mathematical model. Korolko and Sahinoglu [14] solved this problem using the robust optimization method. Zhang and Li [15] handled the charging problem using dynamic programming in the smart grid. Khaki et al. [16] developed a novel hierarchical distributed framework to solve an optimal charging scheduling problem of EVs. Ghofrani et al. [17] used a genetic algorithm to find the appropriate EV charging method. Celli et al. [18] and Yang et al. [19] used a particle swarm optimization algorithm to optimize the charging strategy of the EVs in the power grid. Su et al. [20] proposed a rolling horizon scheduling approach using a genetic algorithm to address the EV charging problem in the energy market. In addition to these studies, various studies have used optimization methods to solve charging scheduling problems, but these approaches may not easily solve the problem because of the complexity of the smart grid [21]. Recently, several researchers have focused on data-based methodologies using machine learning techniques that show powerful data processing and interpretation capabilities. Reinforcement learning is one of the machine learning techniques that has powerful sequential decision-making abilities, as stated in Section 1. Advancing from the reinforcement learning algorithm that stores and updates estimates in a table (e.g., Q-learning algorithm), deep reinforcement learning has a structure that updates values using a neural network. Deep reinforcement learning is suitable for dealing with problems that are complex and have large-scale state and action spaces by using a method that approximates a policy function and value function [22]. Many attempts have been made to optimize the sequential decision making process by applying deep reinforcement learning to the power control system in the smart grid using these characteristics. Liu et al. [23] solved the scheduling problem of a HEMS combined with a PV using deep reinforcement learning. Several approaches have used deep reinforcement learning on EVs in smart grid environments. Lu and Hong [24] proposed a novel demand response algorithm in a smart grid using reinforcement learning and a deep neural network. Wang et al. [25] solved the EV charging scheduling problem for EVs visiting a charging station using the SARSA algorithm. Xiong et al. [26] proposed reinforcement learning based real-time energy management to achieve an optimal power distribution method in an electric vehicle. In addition, there were some studies about the model-free approach of EV charging that learns the time series flow of electricity prices by combining a long short-term memory (LSTM) network with a deep neural network [5,27].

In order to cope with the complex system of the smart grid, a few researchers considered MADRL. MADRL is a study of multiple artificial agents that work in a shared environment, cooperating or competing with the other agents [28]. MADRL can be divided into two methods: Q-learning based and policy-gradient based. QMIX is a MADRL algorithm based on the Q-learning method [29]. QMIX updates Q values using a mixing network. MADDPG and COMA algorithms used in this paper are MADRL algorithms based on the policy gradient method [30,31]. Detailed descriptions of the MADDPG and COMA algorithms will be dealt with in Section 4.

Several studies have focused on applying MADRL algorithms to the power grid environment. Xu et al. [21] solved the problem of managing

**Table 2**
Comparisons of this paper and previous studies.

| Authors (year) | EV charging | PV/ESS | Commuting behavior | Multi-agent | Solution methodology |
|---|---|---|---|---|---|
| Ghofrani et al. [17] | ✓ | | | | Genetic algorithm |
| Zhang and Li [15] | ✓ | | | | Dynamic programming |
| Liu et al. [23] | | ✓ | | | DQN, DDQN |
| Li et al. [36] | ✓ | | | | DDPG |
| Wan et al. [5] | ✓ | ✓ | ✓ | | DQN + LSTM |
| Wang et al. [25] | ✓ | | ✓ | | SARSA |
| Shin et al. [32] | | ✓ | | ✓ | commNet |
| Zhang et al. [27] | ✓ | | ✓ | | CDDPG + LSTM |
| Da Silva et al. [35] | ✓ | | | ✓ | MASCO |
| This paper | ✓ | ✓ | ✓ | ✓ | COMA |

electric devices used at home, including EVs, using MADRL. Shin et al. [32] set the EV charging stations as agents to learn cooperative charging scheduling between the other charging stations to minimize the overall operating cost. Zhang et al. [33] used MADRL algorithms to address the charging scheduling problem in a smart grid. The difference between the paper and this study is that while [33] focused on the energy distribution strategy of the charging stations, this study focuses on the charging scheduling method of the EVs visiting a station. Zhang et al. [34] addressed selecting appropriate EV charging spots using MADRL. The paper by Da Silva et al. [35] is similar to this paper in that EVs are charged using MADRL. However, that paper did not consider the commuting behavior of EVs and the smart grid environment. By factoring in commuting behavior, our research addresses more uncertain and practical charging situations of EV users. Furthermore, this paper deals with an EV charging station in the smart grid, which includes various power systems to charge multiple EVs efficiently.

The differences between previous studies and this paper are depicted in Table 2. In order to consider the commuting behavior of EVs, the situation in which EVs visit charging stations at uncertain times was considered. In addtion, we considered the EV charging station in the smart grid, including the ESS and the PV. Furthermore, by setting up several EVs as agents, we used a MADRL approach to find the optimal decision-making process for each time step.

## 3. Mathematical formulation

In this Section, we introduce the charging scheduling problem we are addressing. Mathematical formulations and the environment of the problem are presented to model and solve the problem. We used the Markov decision process to model the sequential decision-making process of a charging scheduling problem.

### 3.1. Description of EV charging station in the smart grid

In this section, we address a real-time charging scheduling problem of multiple EVs visiting a single EV charging station. Fig. 1 shows the hierarchical structure of an EV charging station in the smart grid. In the EV charging station, there exists PV, which generates electricity from the sun, and ESS, which stores and distributes electricity. EVs visiting the EV charging station receive or return electricity from the ESS. Outside the EV charging station exists, an electricity corporation, called the service provider, which offers electricity to the charging station. A service provider sells electricity to the EV charging station at a real-time electricity price. The EV charging station stores electricity purchased from service providers in an ESS. Electricity stored in the storage system is distributed to EVs in need of charging. There is no cost for the electricity when it receives electricity from the PV inside the charging station. Therefore, in this paper, EV charging stations first use electricity supplied from PV, and then purchase electricity from a service provider for unsatisfied electricity demand.

As the EVs get charged or discharged during the process, both ESS and EVs send and receive electricity while satisfying their capacity constraints. A detailed description of the constraints is given in Section 3.2.

### 3.2. Markov decision process and mathematical formulations

In order to model the system, we define notations and introduce necessary formulations. $E = \{1, 2, \dots, N\}$ represents a set of $N$ EV agents. We define $i \in E$ to represent each EV. The problem occurs within a finite time, $T$, per day, so time, $t$ has a value of $t = 0, 1, 2, \dots, T$

As stated in Section 1, reinforcement learning can be formalized using the Markov decision process. The states, actions, and rewards established to design the EV charging environment are as follows. Equations and accompanying explanations are added to indicate constraints that occur in the environment.

1. State : The state of EV charging problem includes the price state, PV state, ESS state, and the EV state. The description of the state components is as follows. $D = \left\{d_t^0, d_t^1, \dots, d_t^i, \dots, d_t^{N-1}\right\}$ represents the set of remaining charge of the EV $i$ at time step t. The remaining time until departure of EV $i$ is denoted by $RT = \left\{rt_t^0, rt_t^1, \dots, rt_t^i \dots, rt_t^{N-1}\right\}$. The ESS state-of-charge (SOC) at time step t is represented as $ess_t$. The price of electricity at time step t is stated as $p_t$, and $\left(p_{t-23}, \dots, p_t\right)$ refers to the price of electricity in the past 24 h. In addition, $pv_t$ represents the amount of electricity generated from PV at time step, t; and, $\left(pv_{t-23}, \dots, pv_t\right)$ denotes the generated electricity from PV over the last 24 h. $S = \left\{s_t^0, s_t^1, \dots, s_t^i \dots, s_t^{N-1}\right\}$ represents the set of states of EV, $i$, at time step, $t$. Here, $s_t^i = \left[d_t^i, rt_t^i, l_t^i, ess_t, p_{t-23}, \dots, p_t, pv_{t-23}, \dots, pv_t\right]$ denotes the state of EV $i$ at time step $t$.

2. Action : We set the action of EV $i$ at time step $t$ as $a_t^i$. EVs can choose between charging or discharging action, each with a positive or negative sign. EVs can also choose a non-charging action, taking neither charging nor discharging actions. In reality, most EV charging stations provide a discrete charging system. The discrete charging system is a system that charges the EV with a fixed amount of charge each hour. Therefore, in this paper, we set the charging or discharging action as a discrete value [37]. Constraint (1) represents the constraint that the total charging and discharging actions of EVs cannot exceed the possible charging or discharging amount of power allowed at the charging station. Constraint (2) denotes the capacity constraint of the ESS. The sum of the ESS SOC at the previous time step, the sum of all actions of EVs, and the power generated from PV should be between the maximum and minimum capacity of ESS. Constraint (3) represents the capacity constraint of the EV. All the constraints mentioned above are handled in the action selection phase of the algorithm. The actions that violate the constraint are eliminated from the possible action candidates. A detailed implementation of the constraints in the algorithm is illustrated in Section 4.

**Sets**

| | |
|---|---|
| $S$ | Set of observation states |
| $A$ | Set of actions |
| $E$ | Set of EVs |
| $D$ | Set of unsatisfied demand of EVs |
| $U$ | Set of remaining time until departure of EVs |

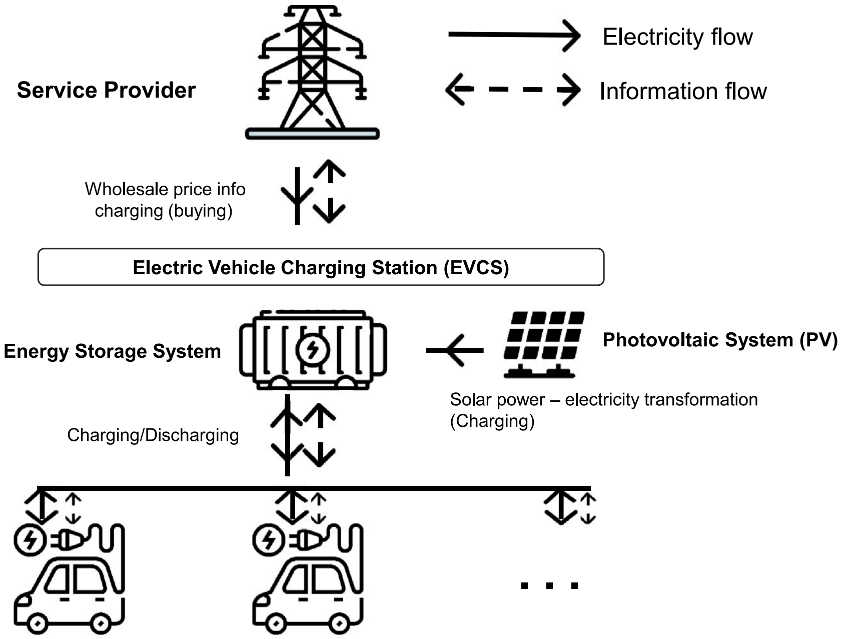

**Fig. 1.** EV charging station structure within the smart grid.

**Parameters**

| | |
|---|---|
| $N$ | Number of EVs |
| $p_t$ | Electricity price at time $t$ |
| $pv_t$ | Photovoltaic generation at time $t$ |
| $soc_t^i$ | State of charge of EV $i$ at time $t$ |
| $r_t^i$ | Reward of EV $i$ at time $t$ |
| $t_{arr}^i$ | Arrival time of EV i |
| $t_{dep}^i$ | Departure time of EV i |
| $l_t^i$ | Position of EV i at time t |
| $ess_t$ | State of charge of ess at time t |
| $ess^{max}$ | Maximum electricity capacity of ESS |
| $ess^{min}$ | Minimum electricity capacity of ESS |
| $ev^{max}$ | Maximum electricity capacity of EV |
| $EC_{ch}^{max}$ | Maximum total charging rate of the charging station |
| $EC_{dch}^{max}$ | Maximum total discharging rate of the charging station |
| $C^B$ | Battery cost of EV |
| $\beta$ | Coefficient of user discomfort |
| $m$ | Depreciation rate of the battery price |

$$-EC_{disc}^{max} \leq \sum_{i=1}^{N} a_t^i \leq EC_{cha}^{max}, \qquad \forall t \in T \tag{1}$$

$$ESS^{min} \leq ess_{t-1} + \sum_{i=1}^{N} a_t^i + pv_t \leq ESS^{max}, \qquad \forall t \in T \tag{2}$$

$$0 \leq soc_{t-1}^t + a_t^i \leq ev^{max}, \qquad \forall t \in T, \forall i \in E \tag{3}$$

3. Reward : The reward of the EV at time, $t$, was configured to satisfy both EV users and EV charging station operators. The first term of Eq. (4) means that the amount of electricity needed for action is purchased at a price, $p_t$, at time step, $t$.

$$r_t^i = \begin{cases} -a_t^i \cdot p_t - C_{a_t^i} - \beta \cdot (d_t^i)^2 & \text{if } t = t_{dep}^i \\ -a_t^i \cdot p_t - C_{a_t^i} & \text{otherwise} \end{cases} \tag{4}$$

The charging system dealt with in this paper is an EV capable of both charging and discharging, in which the EV can be used as an energy storage device. Accordingly, indiscriminate charging and discharging actions for EVs may be made to minimize the operating costs of EV charging stations. Because this can be a significant load on the battery in actual EV operation, we added battery degradation cost to prevent indiscriminate actions. $C_{(a_t^i)}$ represents the battery degradation cost for the action taken by the EV, $i$. The battery degradation cost is defined in Eq. (5) below, with the battery cost, $C^B$, the ratio of depreciation to the economic value of the battery $m$, and the maximum capacity of the EV, $ev^{max}$ [38].

$$C_{a_t^i} = C^B \cdot \left| \frac{m}{100} \right| \cdot \frac{a_t^i}{ev^{max}} \tag{5}$$

In order to charge the EV to the user's desired amount, a strong negative reward is given when the EV is not fully charged at the time, $t_{dep}^i$ when the EV leaves the charging station. The dissatisfaction cost incurred for not satisfying the user's needs can be expressed in the form of a quadratic function as $\beta \cdot (d_t^i)^2$ [39]. The coefficient, $\beta$, suggests a trade-off between satisfaction level and electricity cost, and it will be verified through experiments in Section 5.2.

4. Transition : The transition rule $f$ will transit the $s_t^i$ to the next state $s_{t+1}^i$ based on the action taken by each agent $i$. The transition dynamic can be represented as $s_{t+1}^i = f(s_t^i, a_t^i)$. The dynamics of the state components are updated as follows:

$$d_{t+1}^i = \begin{cases} d_t^i + a_t^i & \text{if } t_{arr}^i \leq t \leq t_{dep}^i \\ d_t^i & \text{otherwise} \end{cases} \tag{6}$$

$$rt_{t+1}^i = \begin{cases} rt_t^i - 1 & \text{if } t \leq t_{dep}^i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$l_t^i$ which represents the position of EV will retain 1 when an EV is at the charging station and 0 otherwise.

$$l_{t+1}^i = \begin{cases} 1 & \text{if } t_{arr}^i \leq t \leq t_{dep}^i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

### 3.3. Environment modeling

We designed the EV charging environment to demonstrate the situation in which reinforcement learning works. Fig. 2 represents the structure of the environment. As the vehicles' arrival and departure times are non-deterministic, the environment initializes the simulation with the beginning of the day and terminates when all EVs leave the charging station.

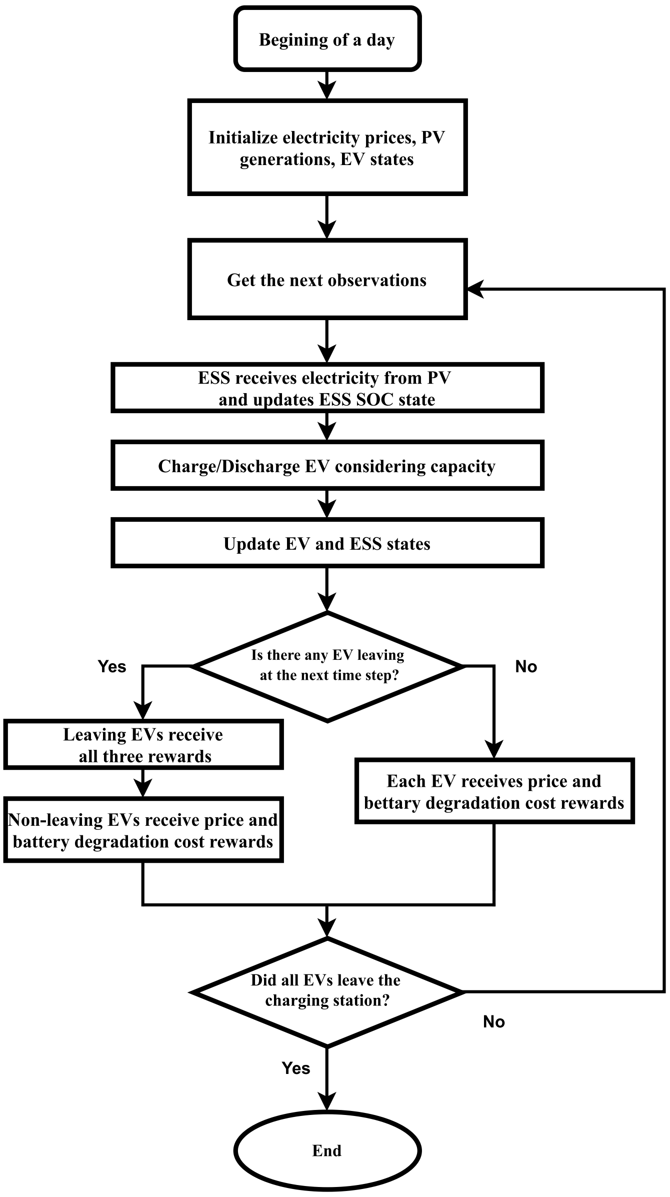
Begining of a day
Initialize electricity prices, PV generations, EV states
Get the next observations
ESS receives electricity from PV and updates ESS SOC state
Charge/Discharge EV considering capacity
Update EV and ESS states
Is there any EV leaving at the next time step?
Yes
No
Leaving EVs receive all three rewards
Non-leaving EVs receive price and battery degradation cost rewards
Each EV receives price and bettary degradation cost rewards
Did all EVs leave the charging station?
No
Yes
End

**Fig. 2.** Structure of the environment.

The environment is initialized with the newly created PV generations, electricity prices, and EV states. EV states consist of EVs' arrival and departure times, the SOC, and the position of the EV. We defined the time step as an hour, to cope with the charging interval term. The ESS receives the solar power generations from the PV, which is within the charging station. After receiving the power from the PV, the ESS updates the SOC, considering the capacity. The ESS distributes the electricity to each EV, and the EV updates its states accordingly. If the power level state of ESS reaches a safety level, it directly draws electricity from the grid. At the charging stage, the EV can be charged only up to the maximum capacity of its battery. Rewards are given concerning the position in the next step of the EV. If EVs leave the next step, they receive all three rewards from the environment. Non-leaving EVs left in the station receive battery degradation and electricity price costs. On the other hand, if all EVs remain at the charging station in the next step, EVs receive the same rewards as the non-leaving EVs in the previous case (i.e., battery degradation cost and electricity price cost). After each EV receives the rewards, all the EV states are updated.

From “beginning of a day” to the time at which all EVs leave the charging station becomes one episode. That is, the terminal state of each episode depends on the position of all EVs. If there are no EVs left in the charging station in the next step, the episode ends. However, if there are any EVs left in the station, the simulation moves to the next step and iterates the process.

## 4. Reinforcement learning algorithms

In this Section, we present MADRL algorithms for solving the EV charging scheduling problem. We approached the problem with the COMA algorithm and the MADDPG algorithm. Section 4.1 provides explanations of the deep reinforcement learning algorithm. Section 4.2 presents the framework of MADRL. Sections 4.3 and 4.4 provide the structure and procedure of both the COMA and MADDPG algorithms.

### 4.1. Deep reinforcement learning algorithm

The Q-learning algorithm, which is one of the most widely used learning methods in reinforcement learning, updates the Q value of the state–action pair, as in Eq. (9).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \tag{9}$$

However, when the dimension of the state corresponding to the input value or the dimension corresponding to the action increases, there exists a limitation to updating with the Q-table. According to Mnih et al. [40], deep Q-networks can quickly and effectively solve problems with high dimension states and actions. If the Q function is approximated as $Q(s, a, \theta) \approx Q_\pi(s, a)$ using the parameter $\theta$, the value function estimation can be made more effectively.

$$L(\theta) = \mathbb{E}\left[\left(R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta') - Q(s_t, a_t; \theta)\right)^2\right] \tag{10}$$

Using the stochastic gradient descent method, it is possible to find the parameter $\theta$ that minimizes the value of Eq. (10).

However, with the Q-learning method, there is a disadvantage in that the policy can change significantly, even with a small change in the Q-value. The policy gradient method emerged to overcome the above shortcoming. The policy gradient method is a method that maximizes reward by updating the policy composed of parameter $\theta$. The policy gradient theorem is shown as Eq. (11).

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim p_\theta(\tau)}\left[\left(\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t)\right)\left(\sum_{t=0}^{T} r(s_t, a_t)\right)\right] \tag{11}$$

The REINFORCE algorithm is the representative algorithm of the policy gradient method. The REINFORCE algorithm updates the parameter, $\theta$, using the estimated gradient in Eq. (11). The process of updating $\theta$ is performed through gradient ascent as $\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$, where $\alpha$ is the learning rate.

### 4.2. Multi-agent deep reinforcement learning

Rather than one agent operating in one environment, it is more common for multiple agents to collaborate or compete in a real environment, such as in sports [41] or in an autonomous vehicle system [42]. MADRL is a method that can train multiple agents in an environment in which agents do not know all information about other agents during the changing environment [43]. In MADRL, as shown in Fig. 3, multiple agents interact with the environment and learn their policies.

Based on the received action, the environment returns the corresponding state and reward. The reinforcement learning methods with independent learners (e.g., independent Q-learning) that were at-

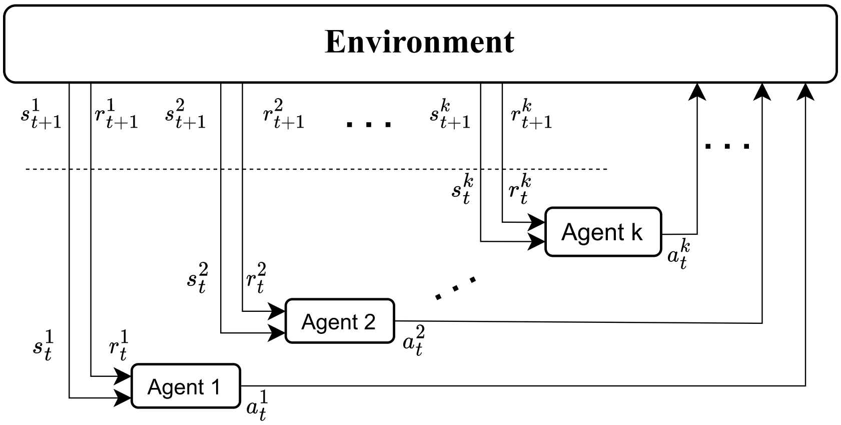

**Fig. 3.** Multiple agents interacting with the environment.

tempted before the MADRL had difficulty handling the non-stationary and partially observable issues. Non-stationary issue is a problem that occurs because the results of an agent's actions are not constant due to the effect of other agents in a given environment. Partially observable issue occurs when one agent does not know all the information about the environment, including information about other agents in the environment. By successfully solving non-stationary and partial observable issues, MADRL can derive optimal decision making for multiple agents.

A variety of learning structures has been proposed for MADRL. In a fully centralized method that shares all the states and rewards of agents, the joint action space expands exponentially. Not only is this a considerable limitation in scalability, but it is not a guaranteed way to find an optimal solution. On the other hand, when agents learn in a fully decentralized way, there is a disadvantage in that it is difficult to consider the cooperative or competitive situation [44]. The centralized training and decentralized execution (CTDE) method is a method that combines the previous two methods. CTDE uses global information during centralized training, and only the agent's local information is adopted to execute the learned policies in a decentralized way [45]. Through this procedure, the CTDE method adequately handles the cooperative situation and guarantees the efficiency of the search space. Therefore, we approached the EV charging scheduling problem with MADRL using the CTDE method.

### 4.3. Counterfactual multi-agent policy gradient algorithm

The theoretical advantages of CTDE have led to the development of numerous MADRL algorithms. COMA, one of the representative algorithms of CTDE, performs centralized training through the critic network, and performs decentralized execution through the actor network. The COMA network uses the credit assignment method in which each agent is assigned a contribution to the overall reward using the actor–critic policy gradient method. Throughout the assignment method, it is known that the COMA algorithm shows good performance in the cooperative situation [31]. The COMA algorithm used a counterfactual baseline for credit assignment. Eq. (12) shows a reward representing the degree of contribution of the agent to the overall reward.

$$D^a = r(s,u) - r(s,(u^{-a},c^a)) \tag{12}$$

In this Equation, $u$ is the joint action space of the agents, $u^{-a}$ represents the joint action space of the other agents except for agent $a$, and $c^a$ means the action independent of the agent's policy. COMA tried to increase the computational efficiency by estimating the Q value using a centralized critic. The advantage function used to calculate the baseline is shown below.

$$A^a(s,u) = Q(s,u) - \sum\nolimits_{u'^a} \pi^a(u'^a \mid \tau^a)\, Q(s,(u^{-a},u'^a)) \tag{13}$$

With the strength in the cooperative situation due to the credit assignment method and the network structure that can handle discrete action space, the COMA algorithm is a reasonable method for the problem we have. Therefore, the scheduling method for charging multiple EVs in the smart grid using the COMA algorithm proposed in this paper is as shown in Algorithm 1 below. We use actor, critic, and target critic networks to implement the COMA algorithm. The procedure of the algorithm is as follows.

**Algorithm 1** COMA algorithm for EV charging scheduling

```
*Initialize* the critic and actor networks $\theta^c$, $\theta^\pi$
*Initialize* the target critic networks $\hat{\theta}^c$
for each training episode $e$ do
    Clear buffer
    for $e_b$ =1 to $Batchsize/n$ do
        *Initialize* electricity price, PV generation, EV states
        for timestep t=1 to T do
            for each EV agent a do
                $h^a_t$ = Actor($o^a_t$, $h^a_{t-1}$, ,$u^a_{t-1}$, $a, u$; $\theta^i$)
                Mask the invalid actions and receive $u^a_t$ from $\pi(h^a_t, \epsilon)$
            end
            Get next state $s_{t+1}$ and reward $r_t$
        end
        Stack memory to buffer
    end
    for t=1 to T do
        Unroll batch using states, actions and rewards
        Calculate targets $y^a_t$ using $\hat{\theta}^c_i$
    end
    for t=T down to 1 do
        $\triangle Q^a_t = y^a_t - Q(s^a_j, u)$
        $\theta^c = \nabla_{\theta^c}(\triangle Q^a_t)^2$
        $\theta^c_{i+1} = \theta^c_i - \alpha \triangle \theta^c_i$
        Every C steps set $\hat{\theta}^c_i = \theta^c_i$
    end
    for t=T down to 1 do
        Calculate $A^a(s^a_t, u)$ based on Eq. (13)
        $\triangle \theta^\pi = \triangle \theta^\pi + \nabla_{\theta^\pi} \log \pi(u|h^a_t) A^a(s^a_t, u)$
    end
    $\theta^\pi_{i+1} = \theta^\pi_i + \alpha \triangle \theta^\pi_i$
end
```

(1) Initialize actor and critic networks for activating and evaluating the action of the agents
(2) The target network of the critic network is initialized for estimating the critic value.
(3) In each episode, $e$, the EV charging environment is created. Mask the invalid actions and select the valid action $u_t^a$ of the EV agent, $a$. Reward, $r_t$, and the next state, $s_{t+1}$, are obtained through this process. Add and gather memories in buffer into single batch.
(4) The TD target, $y_t^a$, is obtained through the target critic, $\hat{\theta}^c$. After finding the critic gradient, we do a critic update. At every C step, the target critic is updated to be the same as the critic.
(5) The advantage function is derived from the counterfactual baseline, as in Eq. (13). Then accumulated actor gradient using the advantage function, and the agent weight is updated.

### 4.4. Multi-agent deep deterministic policy gradient algorithm

The MADDPG algorithm uses the CTDE method with two networks like the COMA algorithm. However, the MADDPG algorithm is different from the COMA algorithm in its number of critics. The COMA algorithm uses one critic network to assign a credit for a reward, but the MADDPG algorithm has multiple critic networks that evaluate each agent. Each critic of the MAPPDG algorithm learns according to the reward structure of the corresponding agent. This characteristic of having multiple critic networks enables MADDPG to handle both cooperative and competitive situations [30].

The detailed explanation of the notation and the equations for the algorithm are as follows. The notations $\mu_k$ and $\mu$ represent the deterministic policy of agent $k$, and the set of continuous policies, $Q_k^\mu$, means the centralized critic for agent $k$ under $\mu$. The observations of all agents are denoted as $s = (o_1, \dots, o_N)$. $M$ denotes the samples from replay buffer, $D$. The centralized actor network is updated as Eq. (14).

$$\nabla_{\theta_k} J(\mu_k) \approx \frac{1}{M} \sum_j \nabla_{\theta_k} \mu_k(o_k^j) \nabla_{a_k} Q_k^\mu(s^j, a_1^j, \dots, a_k^j, \dots, a_N^j)|_{a_k=\mu_k(o_k^j)} \tag{14}$$

Each agent, $k$, learns with all the observations and the actions in the training stage, which alleviates the non-stationary issue. However, in the action selection stage, only the local information of agent, $k$, is used to select the agent's action. With the target value of $y^j$, centralized action–value function $Q_k^\mu$ is updated as Eq. (15).

$$L(\theta_k) = \frac{1}{M} \sum_j \left( y^j - Q_k^\mu(s^j, a_1^j, \dots, a_N^j) \right)^2,$$
$$y^j = r_k^j + \gamma Q_k^{\mu'}(s'^j, a'_1, \dots, a'_N)|_{a'_k=\mu'_k(o_k^j)} \tag{15}$$

The procedure of the MADDPG algorithm is presented in Algorithm 2. The initialization step of the algorithm is similar to Algorithm 1. However, we used $\mathcal{G}$ for exploration of action. We masked the invalid actions which violate the constraints we addressed in Section 3.2. Then, we execute the valid actions and get the next state, $s_{t+1}$, and reward, $r_t$, accordingly. Also, we adopted a replay buffer to the algorithm. Utilizing the replay buffer increases data efficiency, as the algorithm operates in an off-policy method. After storing the samples to the replay buffer, the algorithm updates the actor and critic network based on Eqs. (14) and (15).

**Algorithm 2** MADDPG algorithm for EV charging scheduling

```
$Initialize$ the critic, target critic and actor networks for each EV agent
for each training episode $e$ do
    $Initialize$ electricity price, PV generation, EV states
    $Initialize$ an action exploration process $\mathcal{G}$
    for timestep $t$=1 to $T$ do
        for each EV agent $k$ select action $a_k = \mu_{\theta_k}(o_k)+\mathcal{G}$ from the action space
        Mask the invalid actions and execute feasible actions
        Gather rewards $r$ and next state $s_{t+1}$ according to action $a$
        Save transition $(s, a, r, s_{t+1})$ in replay buffer $D$
        $s \leftarrow s_{t+1}$
        for each EV agent $k$=1 to $K$ do
            Sample a random minibatch of $M$ samples $(s, a, r, s_{t+1})$ from $D$
            Set $y^j = r_k^j + \gamma Q_k^{\mu'}(s'^j, a'_1, ..., a'_K)$
            Update the critic network by minimizing the loss
            $L(\theta_k) = \frac{1}{M}\sum_j \left(y^j - Q_k^\mu(s^j, a_1^j, ..., a_N^j)\right)^2$
            Update the actor network using the policy gradient
            $\nabla_{\theta_k} J \approx \frac{1}{M}\sum_j \nabla_{\theta_k}\mu_k(o_k^j)\nabla_{a_k} Q_k^\mu(s^j, a_1^j, ..., a_k, ..., a_N^j)|_{a_k=\mu_k(o_k^j)}$
        end
        Update target network parameter for each EV agent $k$ :
        $\theta'_k = \tau\theta_k + (1 - \tau\theta'_k)$
    end
end
```

## 5. Computational experiments

In this Section, we conducted an experiment on a system for charging EVs based on the policy learned from the multi-agent deep reinforcement learning algorithm. The experiment was conducted on a Python 3 and Intel Core CPU i5-9400F processor. Through this section, we found a reasonable charging scheduling method and compared it with the existing algorithm to verify its applicability. In Section 5.1, we provide assumptions and system parameters needed for the experiment setup. In Section 5.2, we conduct experiments to analyze the performance of the multi-agent reinforcement algorithms.

### 5.1. Experiment setup

EV users' commuting behavior consists of random variables generated from a truncated normal distribution. The commuting behavior of EV users is adopted in the previous study [46]. The departure time, the arrival time and the SOC of visiting EVs follow a truncated normal distribution to prevent overlap issues that might occur in the environment. The arrival time of EV, $i$, is presented as $t_{arr}^i$. We assumed that EV, $i$, arrives at an average time of 6 p.m. within a range between 3 p.m. and 9 p.m. and follows a truncated normal distribution with a standard deviation of 1. Similarly, we assumed that the departure time of EV, $i$, represented as $t_{dep}^i$, follows a truncated normal distribution and departs between 6 a.m. and 11 a.m., and at an average time of 8 a.m. In Fig. 4, we showed an illustrative example of the commuting behvaior of Evs visiting the charging station. In addtion, we assumed that EVs are, on average, half-charged at this point and can visit with a minimum charge level of 20% and a maximum of charge level of 80%. The SOC of the EV, $i$, at time step, $t$, is represented as $SOC_t^i$. The ESS of a charging station also follows the same assumptions as the SOC of an EV. However, unlike the EV SOC which is determined when the EV arrives at the charging station, the ESS SOC is decided when the environment is initialized. The structure of the artificial neural network used in the experiment is shown in Table 3.

The CTDE algorithms, which show satisfactory performance in the cooperative model, evaluate each EV by a central critic and distribute the policy to implement an action appropriate to their state. In this

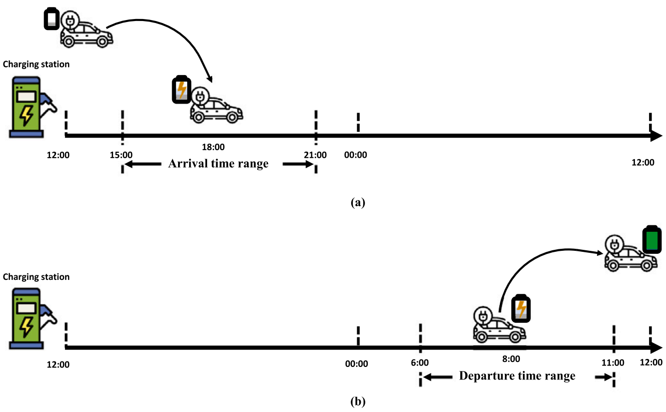

**Fig. 4.** Illustrative example of the commuting behavior of EVs visiting the charging station (a) Arrival times range of EVs that visit the charging station; (b) Departure times range of EVs that have finished charging.

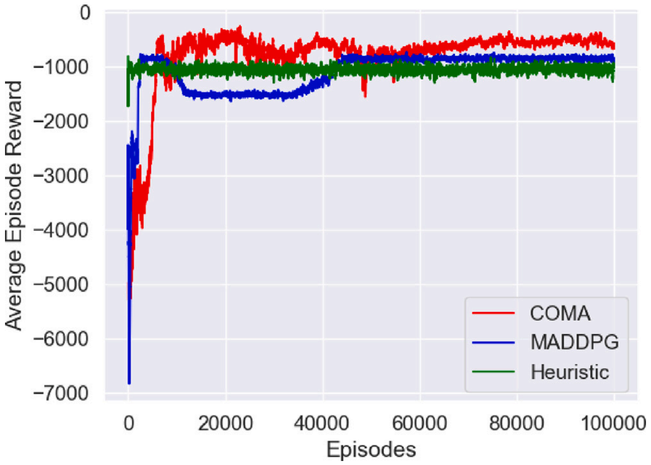

**Fig. 5.** Cumulative rewards during the training process.

**Table 3**
Description of parameters.

| Parameter | Value |
|---|---|
| Episode | 100,000 |
| Time step | 24 |
| Optimizer | adam |
| Discount rate | 0.95 |
| Learning rate | 0.001 |
| Activation function | relu, softmax |

paper, two EVs visiting one EV charging station were selected as agents. The charging and discharging and non-charging action space of each EV is composed of five types of actions, represented as a set $A = \{-25, -12.5, 0, +12.5, +25\}$. We used computational data to calculate the reward and results of the experiment, since it can bring the qualitative measures into results (i.e., user satisfaction). The EV battery capacity, $ev^{max}$, was set to 75 kWh by adopting the commercially available KONA EV model battery. The hourly electricity price data and solar generation data from January 1, 2020, to December 31, 2021, lasting 730 days, were collected from PJM. The input data for representing the state of each agent is normalized for faster training and to avoid being stuck into the local optimum.

### 5.2. Performance evaluation

This section presents the experiment results to examine the performance of our proposed approach. In the experiment, we used the MADDPG algorithm, the COMA algorithm, and the heuristic algorithm. The heuristic algorithm used as a benchmark is constructed by reproducing the algorithm used in [5]. The heuristic algorithm charges EVs preferentially in the evening, when the price is low, and charges the remaining demands during the day. Fig. 5 depicts the training curves of the average of 100 cumulative rewards received by the MADDPG, COMA, and heuristic algorithms.

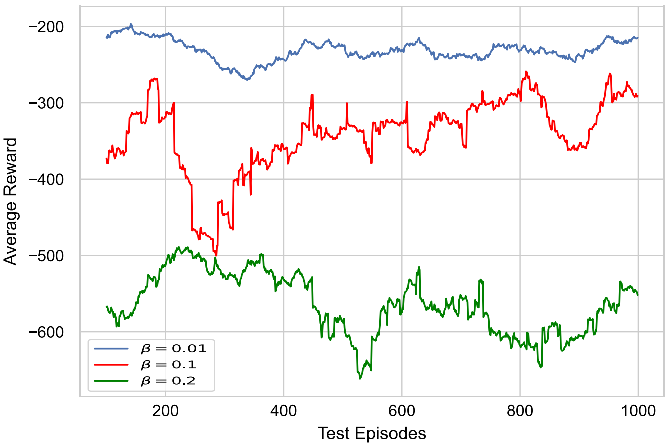

**Fig. 6.** Average rewards under different $\beta$ in the test set.

**Table 4**
Average rewards after convergence.

| | COMA | MADDPG | Heuristic |
|---|---|---|---|
| Average reward | −604.71 | −887.07 | −999.34 |

As shown in Fig. 5, the COMA algorithm, which received a low reward at the beginning of the episode, stably receives a higher reward as the episode proceeds. With the strength stated in Section 4.3, the COMA algorithm shows a stable learning curve when an episode exceeds 20,000 iterations. Table 4 summarizes the average results of the last 10,000 episodes. We proceed with the remaining experiments in this section using the COMA algorithm, which exhibits the highest performance in cumulative rewards.

We conducted a numerical experiment on $\beta$, representing the trade-off between the operating cost and satisfaction level. The validity of the parameter $\beta$ was referred to in previous papers [5,38]. We had set the *beta* value as 0.1 as a default in the experiments. The test data were conducted for 1000 episodes. The penalty coefficients, $\beta$, were chosen to be 0.1, 0.2 and 0.01, respectively, and are indicated by the black, red and green lines.

As shown in the experimental results, Fig. 6 depicts that a better cumulative reward is obtained when the value of $\beta$ shown in black is 0.01. This result stems from the trade-off between the satisfaction level and the operating cost of the penalty coefficient beta. If the penalty coefficient is high, the operating cost of the charging station will increase, but most users have a high level of satisfaction because the demand is satisfied. On the other hand, if the penalty coefficient is low, the operating cost decreases with the unsatisfied demand. Through this, we can confirm that the model's validity is reasonable by matching our intuition, and at the same time, accurately estimating the parameter beta's value is a crucial task that can change the EV charging strategy.

We conducted an experiment on how the agents learn according to different types of hidden layers. We changed the number of hidden layers while keeping the number of nodes the same. The results of the experiment are shown in Fig. 7. We find that as the number of hidden layers increases, the iterations needed for convergence decrease. However, we can also find that too many hidden layers may not get the optimum reward.

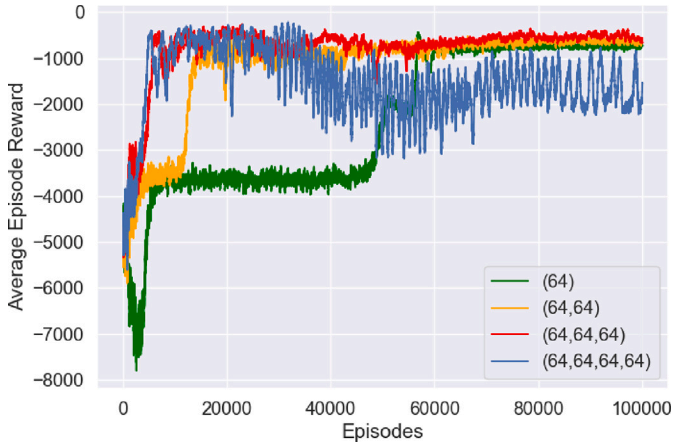

**Fig. 7.** Sensitivity analysis for different hidden layer structures.

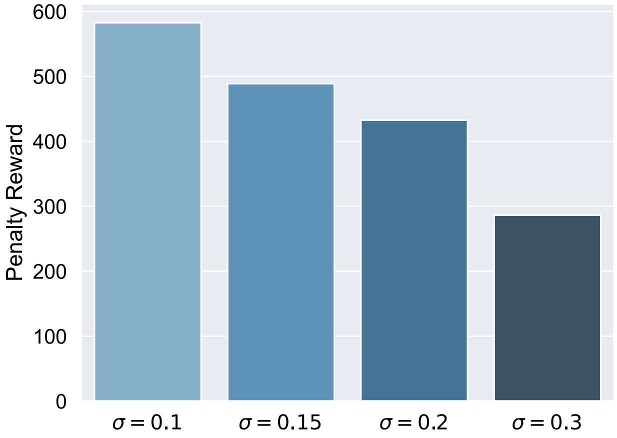

**Fig. 8.** Sensitivity analysis of standard deviation $\sigma$.

### 5.3. Validity discussion

To further validate the EV charging scheduling method, in Fig. 8, the experiment was conducted with the standard deviation value used for estimating the SOC of the visiting EV. Fig. 8 represents the change in total reward with changes in the standard deviation, $\sigma$. The experiment result shows that the penalty reward decreases as the standard deviation value increases. The EV charging station is equipped with a charging system that can sufficiently achieve the target charging demand regardless of the state of the EV. If EVs with a low battery SOC visit, they leave the charging station fully charged, not significantly affecting the reward. However, when vehicles with a large battery SOC visit, the EV can be used as a small ESS that performs charging and discharging actions.

## 6. Conclusions

In this study, we deal with charging EVs visiting an EV charging station in the smart grid that includes a PV and an ESS. The multi-agent deep reinforcement algorithms were used to efficiently charge multiple EVs to minimize the cost incurred in the smart grid. To the best of our knowledge, this document is the first to address the charging scheduling problem of multiple EVs in a smart grid using MADRL. While most of the studies partially alleviated the complex situation to take a mathematical modeling method, we solved the problem without strong assumptions as we formalized the problem to the sequential decision-making problem. The MADDPG and COMA algorithms adopt the most advanced and recent CTDE method to be effective in cooperative situations. Therefore, the algorithms derive the actions of each EV that maximizes the sum of all rewards generated at the EV charging station. The experimental results showed that the charging scheduling method applying the COMA algorithm outperformed the MADDPG algorithm and the existing heuristic algorithm. Future research directions of this paper are as follows.

Including a charging station operator in learning to make pricing decisions might extend the scope of this paper. In this study, we focused on the EV charging scheduling method, but we would like to expand the study to the problem of pricing. Users who visit an EV station purchase electricity at the same price as that sold by the service provider. However, EV charging station operators might think of a problem where charging scheduling and pricing are simultaneously resolved. Through this, we intend to configure a system that can maximize the operating profit of the charging station operator through flexible pricing while maintaining the satisfaction level of EV users. Furthermore, in this paper, the number of EVs is fixed, and the same types of EVs participate. However, we would like to propose a method that can respond flexibly to a more dynamic environment, such as when the number of EVs increases or when other types of EVs visit a charging station. Also, we believe that the charging scheduling method proposed in this paper can be extended to the microgrid field, such as the HEMS within the smart grid.

## CRediT authorship contribution statement

**Keonwoo Park:** Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Investigation, Software. **Ilkyeong Moon:** Conceptualization, Supervision, Validation, Writing – review & editing.

## Data availability

The authors do not have permission to share data.

## Acknowledgments

The authors are grateful for the valuable comments from the associate editor and four anonymous reviewers. This study was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning, South Korea [Grant Number NRF-2019R1A2C2084616].

## References

[1] Makhadmeh SN, Khader AT, Al-Betar MA, Naim S, Abasi AK, Alyasseri ZAA. Optimization methods for power scheduling problems in smart home: Survey. Renew Sustain Energy Rev 2019;115:109362.

[2] Colak I. Introduction to smart grid. In: 2016 international smart grid workshop and certificate program (ISGWCP). IEEE; 2016, p. 1–5.

[3] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89.

[4] Chaudhari K, Ukil A, Kumar KN, Manandhar U, Kollimalla SK. Hybrid optimization for economic deployment of ESS in PV-integrated EV charging stations. IEEE Trans Ind Inf 2017;14(1):106–16.

[5] Wan Z, Li H, He H, Prokhorov D. Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Trans Smart Grid 2018;10(5):5246–57.

[6] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT Press; 2018.

[7] Liang Y, He L, Cao X, Shen Z-J. Stochastic control for smart grid users with flexible demand. IEEE Trans Smart Grid 2013;4(4):2296–308.

[8] Tan KM, Ramachandaramurthy VK, Yong JY. Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques. Renew Sustain Energy Rev 2016;53:720–32.

[9] Liu L, Kong F, Liu X, Peng Y, Wang Q. A review on electric vehicles interacting with renewable energy in smart grid. Renew Sustain Energy Rev 2015;51:648–61.

[10] Kempton W, Udo V, Huber K, Komara K, Letendre S, Baker S, Brunner D, Pearre N. A test of vehicle-to-grid (V2G) for energy storage and frequency regulation in the PJM system. Results Ind-Univ Res Partnersh 2008;32:1–32.

[11] Mersky AC, Samaras C. Environmental and economic trade-offs of city vehicle fleet electrification and photovoltaic installation in the US PJM interconnection. Environ Sci Technol 2019;54(1):380–9.
[12] Wang L, Qin Z, Slangen T, Bauer P, van Wijk T. Grid impact of electric vehicle fast charging stations: Trends, standards, issues and mitigation measures-an overview. IEEE Open J Power Electron 2021;2:56–74.
[13] Hong BK, Kim SH. Recent advances in fuel cell electric vehicle technologies of Hyundai. ECS Trans 2018;86(13):3.
[14] Korolko N, Sahinoglu Z. Robust optimization of EV charging schedules in unregulated electricity markets. IEEE Trans Smart Grid 2015;8(1):149–57.
[15] Zhang L, Li Y. Optimal management for parking-lot electric vehicle charging by two-stage approximate dynamic programming. IEEE Trans Smart Grid 2015;8(4):1722–30.
[16] Khaki B, Chu C, Gadh R. Hierarchical distributed framework for EV charging scheduling using exchange problem. Appl Energy 2019;241:461–71.
[17] Ghofrani M, Arabali A, Etezadi-Amoli M. Electric drive vehicle to grid synergies with large scale wind resources. In: 2012 IEEE power and energy society general meeting. IEEE; 2012, p. 1–6.
[18] Celli G, Ghiani E, Pilo F, Pisano G, Soma GG. Particle swarm optimization for minimizing the burden of electric vehicles in active distribution networks. In: 2012 IEEE power and energy society general meeting. IEEE; 2012, p. 1–7.
[19] Yang J, He L, Fu S. An improved PSO-based charging strategy of electric vehicles in electrical distribution grid. Appl Energy 2014;128:82–92. http://dx.doi.org/10.1016/j.apenergy.2014.04.047, URL: https://www.sciencedirect.com/science/article/pii/S0306261914004000.
[20] Su J, Lie T, Zamora R. A rolling horizon scheduling of aggregated electric vehicles charging under the electricity exchange market. Appl Energy 2020;275:115406. http://dx.doi.org/10.1016/j.apenergy.2020.115406, URL: https://www.sciencedirect.com/science/article/pii/S0306261920309181.
[21] Xu X, Jia Y, Xu Y, Xu Z, Chai S, Lai CS. A multi-agent reinforcement learning-based data-driven method for home energy management. IEEE Trans Smart Grid 2020;11(4):3201–11.
[22] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing atari with deep reinforcement learning. 2013, arXiv preprint arXiv:1312.5602.
[23] Liu Y, Zhang D, Gooi HB. Optimization strategy based on deep reinforcement learning for home energy management. CSEE J Power Energy Syst 2020;6(3):572–82.
[24] Lu R, Hong SH. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. Appl Energy 2019;236:937–49.
[25] Wang S, Bi S, Zhang YA. Reinforcement learning for real-time pricing and scheduling control in EV charging stations. IEEE Trans Ind Inf 2019;17(2):849–59.
[26] Xiong R, Cao J, Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. Appl Energy 2018;211:538–48.
[27] Zhang F, Yang Q, An D. Cddpg: a deep-reinforcement-learning-based approach for electric vehicle charging control. IEEE Internet of Things Journal 2020;8(5):3075–87.
[28] Buşoniu L, Babuška R, De Schutter B. Multi-agent reinforcement learning: An overview. Innov Multi-Agent Syst Appl-1 2010;183–221.
[29] Rashid T, Samvelyan M, Schroeder C, Farquhar G, Foerster J, Whiteson S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: International conference on machine learning. PMLR; 2018, p. 4295–304.
[30] Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I. Multi-agent actor-critic for mixed cooperative-competitive environments. 2017, arXiv preprint arXiv:1706.02275.
[31] Foerster J, Farquhar G, Afouras T, Nardelli N, Whiteson S. Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI conference on artificial intelligence. 2018.
[32] Shin M, Choi D-H, Kim J. Cooperative management for PV/ESS-enabled electric vehicle charging stations: A multiagent deep reinforcement learning approach. IEEE Trans Ind Inf 2019;16(5):3493–503.
[33] Zhang Y, Yang Q, An D, Li D, Wu Z. Multistep multiagent reinforcement learning for optimal energy schedule strategy of charging stations in smart grid. IEEE Trans Cybern 2022.
[34] Zhang W, Liu H, Wang F, Xu T, Xin H, Dou D, Xiong H. Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning. In: Proceedings of the web conference 2021. 2021, p. 1856–67.
[35] Da Silva FL, Nishida CE, Roijers DM, Costa AHR. Coordination of electric vehicle charging through multiagent reinforcement learning. IEEE Trans Smart Grid 2019;11(3):2347–56.
[36] Li S, Hu W, Cao D, Dragičević T, Huang Q, Chen Z, Blaabjerg F. Electric vehicle charging management based on deep reinforcement learning. J Mod Power Syst Clean Energy 2021.
[37] Sun B, Huang Z, Tan X, Tsang DH. Optimal scheduling for electric vehicle charging with discrete charging levels in distribution grid. IEEE Trans Smart Grid 2016;9(2):624–34.
[38] Ortega-Vazquez MA. Optimal scheduling of electric vehicle charging and vehicle-to-grid services at household level including battery degradation and price uncertainty. IET Gener Transm Distrib 2014;8(6):1007–16.
[39] Yu M, Hong SH. Incentive-based demand response considering hierarchical electricity market: A Stackelberg game approach. Appl Energy 2017;203:267–79.
[40] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.
[41] Kalyanakrishnan S, Liu Y, Stone P. Half field offense in RoboCup soccer: A multiagent reinforcement learning case study. In: Robot soccer world cup. Springer; 2006, p. 72–85.
[42] Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving. 2016, arXiv preprint arXiv:1610.03295.
[43] Hu J, Wellman MP, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In: Icml, Vol. 98. Citeseer; 1998, p. 242–50.
[44] Zhang K, Yang Z, Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handb Reinforcement Learn Control 2021;321–84.
[45] Kraemer L, Banerjee B. Multi-agent reinforcement learning as a rehearsal for decentralized planning. Neurocomputing 2016;190:82–94.
[46] Yao L, Lim WH, Tsai TS. A real-time charging scheme for demand response in electric vehicle parking station. IEEE Trans Smart Grid 2016;8(1):52–62.