Competitive Reinforcement Learning for Real-Time Pricing and Scheduling Control in Coupled EV Charging Stations and Power Networks

Adrian-Petru Surani
Cornell Tech, Cornell University
as3259@cornell.edu

Tong Wu
ECE Department, Cornell University
tw385@cornell.edu

Anna Scaglione
ECE Department, Cornell University
as337@cornell.edu

Abstract

This paper proposes a robust Multi-Agent Reinforcement Learning (MARL) approach to optimize the charge schedule and price offered by EV charging stations competing to maximize profits, i.e. the differences between the payments collected by the charging stations and the electricity price set from a distribution system operator. It is assumed that, to prevent energy congestion on the distribution grid, each charging station pays the locational marginal price (LMP) of electricity to serve its customer, determined to be the dual variable of the optimal power flow (OPF) problem. Our proposed RL algorithm trains multiple agents to make optimal charging and pricing decisions at each time step, based solely on past event observations. Additionally, the algorithm takes into account the randomness caused by user behavior, such as travel and wait times, and user flexibility. We observe that, when they are profit maximizing, competing agents vie for higher profits. This intense competition can often lead agents to adopt inefficient policies, mainly due to the disruptions caused by the actions of their competitors. To address this issue, we incorporate constant-sum game theory in the RL policy training. This approach utilizes the minimax policy gradient to maximize the reward of a robust agent, while considering the worst-case scenarios created by competing agents. Simulation results validate that robust agents are capable of generating greater profits than competing agents that do not undergo minimax training and that their presence stabilizes the training.

Keywords: Robust Reinforcement Learning, Twin Delayed DDPG (TD3), EV Charging Pricing.

1. Introduction

The increasing popularity of electric vehicles (EVs) has raised concerns about the additional power grid infrastructure capacity required to serve them, particularly in the case of uncoordinated EV charging [1]. However, EV charging has a lot of spatio-temporal flexibility and the EVs' batteries can also be utilized to provide grid services, potentially making the power grid even more stable and secure through the use of vehicle-to-grid (V2G) technology [2].

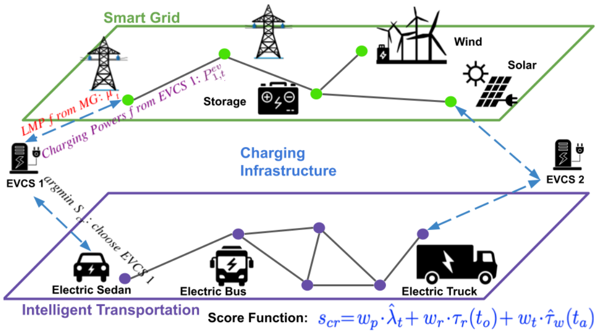

Smart Grid
Wind
Solar
Storage
LMP from MG: μ_t
Charging Powers from EVCS 1: $P^{ev}_{1,t}$
Charging Infrastructure
EVCS 1
EVCS 2
argmin S_{cr}: choose EVCS 1
Electric Sedan
Electric Bus
Electric Truck
Intelligent Transportation
Score Function: $s_{cr} = w_p \cdot \hat{\lambda}_t + w_r \cdot \tau_r(t_o) + w_t \cdot \hat{\tau}_w(t_a)$

Figure 1. Parameters of Users DP function.

Figure 1 depicts the interconnection between transportation networks and power grids, facilitated by V2G technology. This technology enables EVs to return energy to the grid, transforming them from merely shiftable loads into distributed energy sources [3]. One of the challenges is how to harness this flexibility. Dynamic pricing and scheduling algorithms informed by grid congestion have emerged as promising solutions to incentivize peak shaving, or valley filling. Dynamic

This research was supported in part by the DoD-NAVY-ONR #N00014-23-1-2349. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

URI: https://hdl.handle.net/10125/106749
978-0-9981331-7-1

pricing allows charging providers to adapt prices for end-users based on various factors such as demand, time of day, availability of renewable energy sources and price of electricity, which reflects the congestion on the grid. The price posted would motivate end-users to change their electricity consumption decisions in response to financial incentives [4], [5]. In response to a price signal, dynamic scheduling can enhance grid reliability [6], reduce charging operational costs [7], enable auxiliary services [8], and facilitate the integration of renewable generation in commercial microgrids [9], leading to a more efficient and sustainable energy system.

In this work, we aim to solve the Charging Station Dynamic Pricing and Scheduling (CSDPS) problem by MARL in a competitive environment. In our model, each EV selects a station for charging by comparing a score that depends on various factors (e.g. price, wait-time, travelling time to the charging station, etc.) and charging requests may be continuously issued by EV drivers at any time of the day. Several charging stations are available to serve the EVs' demand and attract customers providing a competitive dynamic price for the charge requested and an estimated waiting time, that in turn affects each EV driver's station selection; the dynamic price is chosen to maximize their operational effectiveness from a competitive market of charging stations services. In particular, since the stations' available charging resources are limited, and an EV's charging process may occupy the charging spot for several hours, the charging stations' past pricing incentive and scheduling decisions for a charging request will have long-term effects on the future feasible actions. To develop long-term optimal real-time pricing policies in such a highly dynamic charging market environment, it's intuitive one could apply RL to solve the CSDPS problem, optimizing a long-term business profit goal.

In an urban area with a large number of available charging stations, a centralized RL method that manages the entire system using a single agent would be unrealistic [10] and it is natural to envision market forces at play for matching economically demand and supply. To tackle the CSDPS problem, we present a MARL approach, Minimax Multi-agent Twin Delayed Deep Deterministic Policy (M3TD3), which draws inspiration from the Twin-Delayed Deep Deterministic (TD3) framework introduced in [11] and is inspired by [12]. In our approach each charging station acts as an independent agent. This approach enables the creation of a scalable and efficient distributed multi-agent management system, while also emphasizing the robustness of each agent against perturbations caused by competing agents. Reviews on cooperative and competitive multi-agent learning have addressed these challenges from a theoretical perspective [12]–[14]. Various techniques have been considered to improve the robustness of the main agent, such as robust MARL and adversarial perturbations. Inspired by competitive multi-agent learning methods, this paper makes two contributions.

- Our first contribution is adapting the robust Deep Reinforcement Learning (DRL) called the Minimax Multi-agent Deep Deterministic Policy Gradient (M3DDPG) algorithm proposed in [12] to the CSDPS. We train multiple agents to optimize their charging and pricing decisions at each time step. To address competition between these agents, we have integrated constant-sum game theory with RL policy training. Specifically, we employ the minimax policy gradient technique to maximize the reward of a robust agent, even in the presence of worst-case scenarios created by competing agents.
- We include market grid congestion signals through Distribution Locational Marginal Prices (LMPs) computed by a distribution system operator (DSO) for each of the EV charging stations involved, so as to ensure the high efficiency and reliability of grid operations as they serve the charging station demand. We consider the presence of distributed renewable energy resources and explore how they can reduce the energy costs for the DSO and the EV charging stations operators.

The remainder of this paper is structured as follows. Section 2 introduces the level model for the charging station, and introduces Markov Decision Processes in this context. In Section 3, we model the distribution power level using LMPs for EV charging stations. Section 4 presents the robust M3TD3 reinforcement learning framework for addressing the CSDPS problem. To validate the proposed M3TD3 approach, we present numerical results in Section 5. Finally, Section 6 concludes this paper.

2. Charging Station Agent Model

2.1. System Model

In this study, we examine the operation of multiple EV charging stations agents (EV-CSA) over a time horizon divided into T time slots. EVs generate charging station demand at random times, following a Poisson distribution. We denote by $\mathcal{R}_t$, the set of EVs who just arrived at the charging station, $\mathcal{J}_t$, the set of EVs that are currently waiting to be charged, $\mathcal{I}_t$, the set of EVs who started charging at the beginning of time slot t, and $\mathcal{L}_t$, the set of EVs that are currently being charged. These four sets correspond to queues that are managed on a First In First Out (FIFO) basis. At each time slot t, the EVs that have been served will be departing from the queue $\mathcal{L}_t$. At the same time, when the queue $\mathcal{L}_t$ has empty seats, cars from the waiting set $\mathcal{J}_t$ will move to $\mathcal{L}_t$. The new-arrival EV users $\mathcal{R}_t$ will move to the waiting set $\mathcal{J}_t$. When the chosen charging station is full, EVs will be redirected to another station.

There is also a limit on both the charging set $|\mathcal{L}_t| \leq B_c$ and waiting set $|\mathcal{J}_t| \leq B_w$. For all EVs $i \in \mathcal{I}_t$, let t_i, τ_i, and d_i denote the time at which charging begins (or when the contract is signed), parking time, and charging demand, respectively. In particular, the demand d_i must be satisfied before the departure of EV i at time $t_i + \tau_i$ that is also referred as the deadline. At each slot t, a charging station determines the charging rate of each EV $i \in \mathcal{L}_t$, denoted as x_{it} kWh. The charging rates are constrained by

$$x^{\min} \leq x_{it} \leq x^{\max}, t = 1, \ldots, T, \quad \forall i \in \mathcal{L}_t \tag{1}$$

$$\sum_{t=t_{a,i}}^{t_{a,i}+\tau_i} x_{it} = d_i, \quad \forall i \tag{2}$$

where $x^{\max}$ ($x^{\min}$) is the maximum (minimum) individual charging rate and we will assume no upper bound on the aggregate charging rate due to the limited seats availability. Moreover, Eq. (2) ensures that the charging demand, denoted as d_i, required by the EV user i within the designated time window, starting from its beginning charging time, t_i, and concluding at its departure time, $t_i + \tau_i$.

The CSA sets a dynamic charging price λ_i \$/kWh $\forall i \in \mathcal{I}_t$ at time t for all EVs that start charging then. Prices differ for EVs accepting contracts at varied times, mirroring real scenarios where EV owners agree to listed prices upon beginning to charge. We assume EVs are price sensitive. In response to λ_i, an EV $i \in \mathcal{I}_t$ sets its charging demand as $d_i = D_i(\lambda_i)$ kWh, where $D_i(\cdot)$ is the Price Demand (PD) function shown in Fig. 2. Notably, if charging prices are high, EV users might choose a different station, setting demand to zero. Thus, for an elastic user i, $D_i(\lambda_i)$ is zero whenever λ_i exceeds the threshold $\overline{\lambda}_i$, which can be seen at 0.4, 0.5 and 0.7 \$/kWh. At the same time, an inelastic user will have a constant demand, independent of the price set by the charging station.

Note that in Fig. 2, EV drivers are classified into four different classes, with some being price-sensitive and others not. The sensitivity to pricing is the first source of randomness that must be taken into account by our pricing and scheduling algorithm.

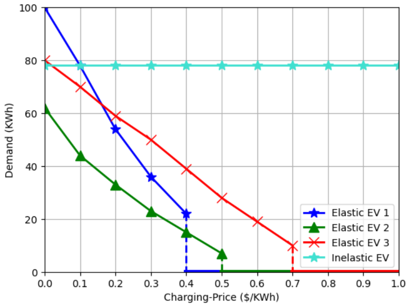


Figure 2. Demand price function for EV users.

2.2. EV User Model: Randomness of Arrivals

We define the time when the EV driver opts to charge their vehicle by t_o, the time when it arrives at the station by t_a, and the time when it begins to charge by:

$$t = \overbrace{t_o + \tau_r(t_o)}^{\triangleq t_a} + \hat{\tau}_w(t_a), \tag{3}$$

where $\tau_r(t_o)$ is the time it will take the EV to reach the CSA if the journey starts at t_o, and $\hat{\tau}_w(t_a)$ is the predicted waiting time at the car's arrival at the EV CSA, t_a. In Fig. 1, the selection of the most cost-effective charging station for EVs can be achieved by defining a score function based on the charging price at time t, the forecasted waiting time $\hat{\tau}_w(t_a)$ of the charging stations, and the travel time to reach each of the stations. Travel times are determined by the distances from an EV driver's starting point to each of the charging stations. The score function can be defined to incorporate these factors:

$$s_{cr} = w_p \cdot \hat{\lambda}_t + w_r \cdot \tau_r(t_o) + w_t \cdot \hat{\tau}_w(t_a), \tag{4}$$

where $\hat{\lambda}_t$ and $\hat{\tau}_w(t_a)$ denotes the forecasted ground-truth price λ_t and forecasted ground-truth waiting time $\tau_w(t_a)$. The weights w_p, w_w, and w_t should be assigned based on the driver's priorities.

To forecast $\hat{\lambda}_t$ and $\hat{\tau}_w(t_a)$ one can use features that describe the current state, such as current prices, waiting times and departures of the charging station and apply a regression algorithm of choice. In this paper we do not focus on optimizing this prediction, but rather on modeling the whole decision process.

2.3. Markov Decision Process

At the start of each time slot t, the CSA sets the charging price and schedule based on observed past and current events such as EV charging demand, departure times, and electricity prices. This decision influences future charging demands. Therefore, the optimal choice is an MDP solution. For clarity in this subsection, we'll use simplified notations for a single agent, eliminating the agent index k:

- S_t denotes the CSA's system state at time t.
- A_t denotes the CSA's action at time t.
- r_t denotes the CSA's reward function at time t.

System State: The state at time t is described by:

$$S_t = \Big(\{\mathcal{J}_t, \mathcal{L}_t\}, \{(\tilde{d}_i^t, \tilde{\tau}_i^t, \lambda_i), \forall i \in \mathcal{L}_t\}\Big), \tag{5}$$

where $\tilde{d}_i^t$ and $\tilde{\tau}_i^t$ are the residual charging demand and parking time of EV i at time t.

Action and Transition Function: Based on S_t and the observation of the real-time electricity price μ_t, the charging station decides the charging price λ_t to the EV drivers $\mathcal{I}_t$ and the charging rate of each EV x_{it} for the charging EV seats $i \in \mathcal{L}_t$. As such, the action A_t at time t is described by a high-dimensional vector, i.e.,

$$A_t = (\{\lambda_i, \forall i \in \mathcal{I}_t\}, \{x_{i,t}, \forall i \in \mathcal{L}_t\}) \tag{6}$$

Under the charging schedule x_{it}, we have

$$\tilde{d}_i^{t+1} = \tilde{d}_i^t - x_{it}, \quad \forall i \in \mathcal{L}_t \tag{7}$$

at the beginning of time slot $t+1$ and the initial residual charging demand is $X_i = D_i(\lambda_i)$.

Meanwhile, the residual parking time decreases as time increases from t to $t+1$, i.e.,

$$\tilde{\tau}_i^{t+1} = \tilde{\tau}_i^t - 1, \ \forall i \in \mathcal{L}_t \tag{8}$$

where $\tilde{\tau}_i^{t+1} = 0$ or $\tilde{d}_i^{t+1} = 0$ indicates that EV i departs. Thus, when the charging commences at time t, the car will continue to charge either until $t + \tau_i$ or until the remaining charging demand is fully met, whichever occurs first. A higher value of $x^{\min}$ guarantees that the initial charging demand can be adequately met within the specified charging deadline, $t + \tau_i$. The state transition function is referred to with the following notation:

$$S_{t+1} := \mathcal{T}(S_t, A_t, \mathcal{I}_t) \tag{9}$$

where μ_t represents LMP at time t.

Reward Function and Decision Problem: The reward function is the profit of the charging station, the benefit of EV customers, the social welfare, etc. Without loss of generality, we suppose that the objective is to maximize the profit of the charging station. The reward function observed by the charging station at time t is the difference between the payment it collects and the electricity bill it pays. That is,

$$r_t(S_t, A_t) := \sum_{i \in \mathcal{I}_t} \lambda_i D_i(\lambda_i) - \mu_t \overbrace{\sum_{i \in \mathcal{L}_t} x_{it}}^{P_t^{ev}} \tag{10}$$

where μ_t denotes the LMP, while P_t^{ev} represents the electric demand to charge the vehicles, that couples the EV charging stations decisions drawing power from the same distribution network.

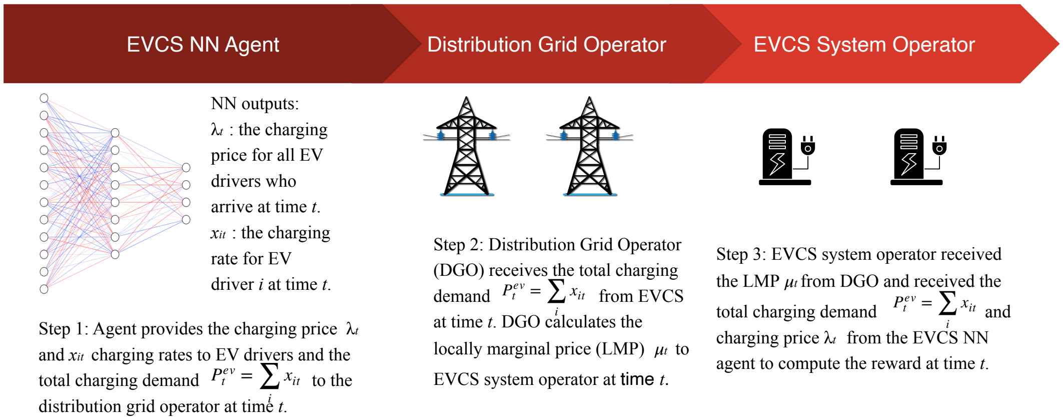
3. Power Network's Level Model - Multiagents

In this section, we will discuss multiple agents that represent various EV charging stations, coupled with distribution power grids. The distribution power grid can be depicted as a graph $G(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ denotes the set of buses, and $\mathcal{E}$ denotes the set of distribution lines. The set of EV charging stations connected to the grid forms a subset of $\mathcal{N}$, denoted by $\mathcal{K} \subseteq \mathcal{N}$ with cardinality $|\mathcal{K}| = K$.

As illustrated in Fig. 1, the dynamic pricing from the power grid is important to maintain power grid stability and avoid demand congestion. For instance, if an EV charging station agent (EV CSA) $k \in \mathcal{K}$ sets a lower price, it may attract more customers and increase the bus congestion in the vicinity of EV CSA $k \in \mathcal{K}$. Therefore, it becomes imperative to consider power flow constraints. Locational marginal prices (LMP) can signal EV CSA $k \in \mathcal{K}$ about the tightness of such constraints on a certain bus. Each CSA will have to charge higher prices due to high electricity costs and, in turn, this will divert EV users to other charging stations. Fig. 3 depicts the workflow connecting the EV-CSAs and the distribution grid, illustrating how the LMP links the DSO and EV-CSAs at time t.

To model power flow constraint congestion, we utilize the AC OPF problem to calculate the LMPs for each bus that connects a charging stations in the set $\mathcal{K} \subseteq \mathcal{N}$, where $\mathcal{N}$ is the complete set of buses in the grid. Let Y be the nodal admittance matrix, where $Y_{nm} = G_{nm} + jB_{nm}$ for line (n, m). Note that $G_{nn} = g_{nn} - \sum_{n \neq m} G_{nm}$ and $B_{nn} = b_{nn} - \sum_{n \neq m} B_{nm}$.

$$\min \sum_{n \in \mathcal{G}} C_n(P_{G_n,t}) + \beta \sum_{n \in \mathcal{N}} \big||V_{n,t}| - v_0\big|_2^2$$



NN outputs:
λ_t : the charging price for all EV drivers who arrive at time t.
x_{it} : the charging rate for EV driver i at time t.

Step 1: Agent provides the charging price λ_t and x_{it} charging rates to EV drivers and the total charging demand $P_t^{ev} = \sum_i x_{it}$ to the distribution grid operator at time t.

Step 2: Distribution Grid Operator (DGO) receives the total charging demand $P_t^{ev} = \sum_i x_{it}$ from EVCS at time t. DGO calculates the locally marginal price (LMP) μ_t to EVCS system operator at time t.

Step 3: EVCS system operator received the LMP μ_t from DGO and received the total charging demand $P_t^{ev} = \sum_i x_{it}$ and charging price λ_t from the EVCS NN agent to compute the reward at time t.

Figure 3. Flow of pricing information: Distribution Network and EV CSA.

$$\text{s.t.} P_{G_n,t} - P^{ev}_{n\in\mathcal{K},t} - P_{n,t} = \sum_{m\in\delta(n)} \Re\left\{Y^*_{nm}V^*_{m,t}V_{n,t}\right\},$$

$$Q_{G_n,t} - Q_{n,t} = \sum_{m\in\delta(n)} \Im\left\{Y^*_{nm}V^*_{m,t}V_{n,t}\right\},$$

$$\underline{P}_{G_n,t} \le P_{G_n,t} \le \overline{P}_{G_n}, \underline{Q}_{G_n,t} \le Q_{G_n,t} \le \overline{Q}_{G_n},$$

$$|S_{nm}| = |V_n(V^*_n - V^*_m)Y^*_{nm}| \le S^{max}_{nm} \tag{11}$$

$$\underline{V_n} \le |V_{n,t}| \le \overline{V_n}, \tag{12}$$

where where $\delta(n)$ is the set of neighboring buses of bus n, $\mathcal{G}$ denotes the set of buses which have generators, $n \in \mathcal{K}$ denotes the EV charging station that connects with the nth bus of the power grid. Our optimization problem involves two objectives. The first objective aims to minimize the fuel costs, while the second objective focuses on maintaining the voltage regulation. To balance these two objectives, we introduce a constant weight β. $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote the real and the imaginary part of a complex number, respectively. $P_{G_n,t}$, $Q_{G_n,t}$, $P_{n,t}$, $Q_{n,t}$ represent the active and reactive power at the bus n. Similarly, $\underline{P}_{G_n}, \underline{Q}_{G_n}$ and $\overline{P}_{G_n}, \overline{Q}_{G_n}$ correspond to the lower bounds and upper bounds for the active and reactive power generation. $|V_n|$ corresponds to the voltage magnitude at bus n, and $\underline{V_n}, \overline{V_n}$ the associated lower and upper bounds. Let v_0 represent the desired voltage magnitude, which is typically set at 1 per unit.

For simplicity, let $\boldsymbol{u}_t$ denote the vector of all control variables, including P_{G_n}, Q_{G_n}, and $\boldsymbol{\varphi}_t$ denote the vector of all state variables, including the voltage magnitude and angle at every bus and phase. We define $H_h(\boldsymbol{\varphi}_t, \boldsymbol{u}_t) \le 0$ to represent the inequality constraints in (12). The Lagrangian function of the formulated AC OPF problem can be written as:

$$Lag(\boldsymbol{\varphi}_t, \boldsymbol{u}_t, \boldsymbol{\mu}_t, \boldsymbol{\nu}_t, \boldsymbol{\rho}_t) = \sum_{G_n\in\mathcal{G}} C_n(P_{G_n,t}) - \sum_{n\in\mathcal{N}} \mu_{n,t}$$
$$\Big(P_{G_n,t} - P^{ev}_{n\in\mathcal{K},t} - P_{n,t} - \sum_{m\in\delta(n)} \Re\{Y^*_{nm}V^*_{m,t}V_{n,t}\}\Big)$$
$$- \sum_{n\in\mathcal{N}} \nu_{n,t}\Big(Q_{G_n,t} - Q_{n,t} - \sum_{m\in\delta(n)} \Im\{Y^*_{nm}V^*_{m,t}V_{n,t}\}\Big)$$
$$+ \sum_{h\in\mathcal{H}} \rho_{h,t}H_h(\boldsymbol{\varphi}_t, \boldsymbol{u}_t) \tag{13}$$

where $\mu_{n,t}$ and $\nu_{n,t}$ are the Lagrange multipliers corresponding to the active power balance equation, and the reactive power balance equation. ρ_h is the Lagrange multiplier associated with the inequality constraint $H_h(\boldsymbol{\varphi}_t, \boldsymbol{u}_t) \le 0$. From the optimal solution $(\boldsymbol{\varphi}^*, \boldsymbol{u}^*)$ of the AC OPF problem in (12), the LMP is:

$$\mu_{n,t} = \frac{\partial Lag}{\partial P^{ev}_{n,t}} = \text{Lagrange Multiplier of bus } n \in \mathcal{K} \tag{14}$$

where $\mu_{n,t}$ represents the locally marginal price of bus $n \in \mathcal{K}$ with an EV charging station connected. Recall that the LMP $\mu_{n,t}$ determines the CSA reward (10).

4. Minimax Multi-Agent Twin Delayed DDPG (M3TD3)

In this section we introduce the instance of M3TD3 algorithm we adapted to solve the CSDPS problem, which is inspired by M3DDPG [12]. This algorithm builds on top of MARL formulations, with the aim of improving the robustness of the learned policies. The M3DDPG approach has two main features. The first one involves incorporating a minimax optimization into the learning objective, drawing inspiration from game theoretical concepts. The second one is using Multi-Agent Adversarial Learning (MAAL) techniques to address computational intractability issues.

4.1. Preliminaries: Twin Delayed Deep Deterministic Policy Gradient (TD3)

The TD3 method utilizes an actor-critic framework, where the policy function's parameters are updated based on an approximate value function, known as the critic [11]. The actor is a policy function π_θ that is parameterized by θ for action selection and the critic is a state-value function Q_ξ, which is parameterized by ξ and provides a critical evaluation of the actor's chosen action. Q-learning utilizes temporal difference learning [15] to learn the value function, based on the Bellman equation [16]. Deep Q-learning involves updating the network through temporal difference learning, using a critic network $Q_\xi(S, A)$ to maintain a fixed objective y over multiple updates:

$$y = r + \gamma Q_\xi(S, A), \;\; A \sim \pi_\phi(A|S), \qquad (15)$$

where actions are picked from a target actor network π_ϕ.

Target Networks: To enhance the stability of deep reinforcement learning and minimize function approximation errors, researchers commonly incorporate *target networks* [11]. In our setup, we utilize two critic networks, Q_{ξ_1} and Q_{ξ_2}, in conjunction with two corresponding target networks, $Q_{\xi'_1}$ and $Q_{\xi'_2}$. The Clipped Double DQN algorithm [11] utilizes the target networks by taking the minimum value estimate between the two, as follows:

$$y = r + \gamma \min\{Q_{\xi'_1}(S, A), Q_{\xi'_2}(S, A)\}. \qquad (16)$$

Critic Networks: The critic networks update their parameters by:

$$\xi_{j=1,2} \leftarrow \arg\min_{\xi_{j=1,2}} \frac{1}{N} \sum (y - Q_{\xi_{j=1,2}}(S, A))^2, \quad (17)$$

where N is the batch size and y is calculated using the target networks as defined in (16). Following the update of the critic networks, the target networks' weights are adjusted at each timestep by a constant factor χ applied to the respective critic network's weights [11]:

$$\xi'_1 \leftarrow \chi\xi_1 + (1 - \chi)\xi'_1, \xi'_2 \leftarrow \chi\xi_2 + (1 - \chi)\xi'_2. \quad (18)$$

where the critic networks with parameters ξ_1 and ξ_2 correspond to Eq. (17), while the target networks with parameters ξ'_1 and ξ'_2 correspond to Eq. (15). To achieve stability and reduce function approximation errors, the target networks and critic networks are alternatively updated by each other [11].

4.2. M3TD3

One of our key contributions is the extension of Minimax Robust Learning, as outlined in [12], to the multiagent TD3 RL framework. One way to expand MDPs to multiple agents is through the use of partially observable Markov games [17]. A Markov game for K agents is defined by a set of states, in our case are $\mathcal{S}$ describing the joint observations $S_1, \cdots, S_K$ of all agents with each agent representing one EV CSA, and $\mathcal{A}$ describing the joint actions $A_1, \cdots, A_K$ of all agents, and $\mathcal{A}_{-k} \triangleq \mathcal{A}/A_k$ describing the set obtained from $\mathcal{A}$ by removing the element A_k. Each agent k uses a stochastic policy $\pi_{\phi_k} : S_k \mapsto A_k$, parameterized by ϕ_k, which produces the next state according to the state transition function $\mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}'$. To extend the notation from a single agent to multiple agents, we can replace the variables S and A with the sets $\mathcal{S}$ and $\mathcal{A}$, respectively. For the kth CSA critic network, we can revise the Bellman equation in (19) as follows:

$$Q^k_\xi(\mathcal{S}, \mathcal{A}) = r_k + \gamma \mathbb{E}[Q^k_\xi(\mathcal{S}', \mathcal{A}')], \; \mathcal{A}' \sim \{\pi_{\phi_k}(A'_k|S'_k)\}. \qquad (19)$$

where r_k represents the reward of CSA k (10). Likewise, the double target networks of agent k in (16) are:

$$y_k = r_k + \gamma \min\{Q^k_{\xi'_1}(\mathcal{S}, \mathcal{A}), Q^k_{\xi'_2}(\mathcal{S}, \mathcal{A})\}. \qquad (20)$$

The parameters of the critic networks can be updated using the same approach as described in (17) and (18). In RL, the objective is to find the optimal policy π_{ϕ_k}, with parameters ϕ_k, which maximizes the expected return $J(\phi_k) = \mathbb{E}_{A_k \sim \pi_{\phi_k}}[r_0]$. Using the Q function defined previously, the gradient of the policy has two parts:

$$\nabla_{\phi_k} J(\phi_k) = \frac{1}{N} \sum \nabla_{A_k} Q^k_{\xi_1}(\mathcal{S}, \mathcal{A}_{-k}, A_k)\Big|_{A_k = \pi_{\phi_k}(S_k)}$$
$$\nabla_{\phi_k} \pi_{\phi_k}(S_k) \qquad (21)$$

where N denotes the training batch size. We can utilize Eq. (21) to maximize the Q function, thereby maximizing the reward. However, the above strategy for agent k is sensitive to other agents' behavior and can easily lead to oscillations in the learning curves. Therefore, we will begin with the following formulation of the minimax learning objective to train a robust agent

k, which is inspired by [12]:

$$\nabla_{\phi_k} J(\phi_k) = \frac{1}{N}\begin{bmatrix} \nabla_{\phi_k}\pi_{\phi_k}(S_k)\,\nabla_{A_k} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}^\star_{-k}, A_k\right) \\ A_k \\ \mathcal{A}^\star_{-k} \end{bmatrix}$$

$$A_k = \pi_{\phi_k}(S_k),$$

$$\mathcal{A}^\star_{-k} = \arg\min_{\mathcal{A}_{-k}} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}_{-k}, A_k\right)$$

The embedded optimization is subsequently modified through the application of Multi-Agent Adversarial Learning (MAAL) [12], which entails replacing the inner-loop minimization process with a single-step gradient descent. This minimization considers the the worst-case scenarios created by competing agents and leads to the revised equation:

$$\nabla_{\phi_k} J(\phi_k) =$$

$$\frac{1}{N}\begin{bmatrix} \nabla_{\phi_k}\pi_{\phi_k}(S_k)\,\nabla_{A_k} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}^\star_{-k}, A_k\right) \\ A_k \\ A^\star_{-k} = A_{-k} - \alpha_{-k}\nabla_{A_{-k}} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}_{-k}, A_k\right) \end{bmatrix},$$

$$A_k = \pi_{\phi_k}(S_k)$$

$$\mathcal{A}^\star_{-k} = \arg\min_{\mathcal{A}_{-k}} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}_{-k}, A_k\right) \qquad (22)$$

where $\alpha_1, ..., \alpha_K$ represent the gradient step sizes. In the extreme scenario where $\alpha = 0$, the M3TD3 method reverts to the original MATD3 algorithm. Conversely, as α increases, the policy learnt becomes more robust, but the optimization grows more challenging. The generative adversarial network proposes calculating the gradient descent with a fixed norm, specifically $g = \nabla_x f_\theta(x; y)$ where x signifies the classifier's input data, and y represents the label [18]. Consequently, within our M3TD3 algorithm, we can adaptively determine the gradient by

$$g_k = \frac{\nabla_{A_k} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}_{-k}, A_k\right)}{\left\|\nabla_{A_k} Q^k_{\xi_1}\left(\mathcal{S}, \mathcal{A}_{-k}, A_k\right)\right\|}. \qquad (23)$$

5. Experimental Results

In this section, we present the results of our numerical experiments. In Fig. 4, LMPs were calculated considering two charging stations connected to the IEEE 18-bus distribution network **at buses 3 and 8** for the 2-station case, and **at buses 3, 8 and 9** for the 3-station case. The active and reactive demands are scaled from the Texas time-series datasets [1]. The PYPOWER 5.1.16 is utilized to compute the LMP for each EV CSA. To simplify notation throughout the following sections and figures, 1 *iteration* constitutes 2000 *updates* to the model's parameters, and we average over 500 updates. In the case of 2 agents, the training process is run for 300 iterations and the testing for 100 iterations, while in the 3-agents case the testing phase is run for 40 iterations.

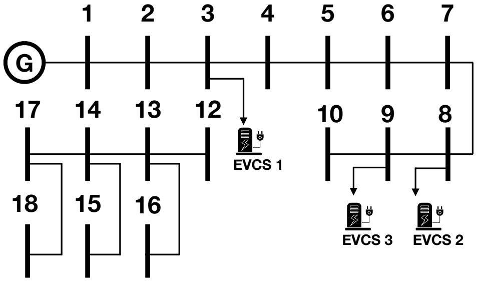


Figure 4. IEEE 18-bus system with EV CSAs.

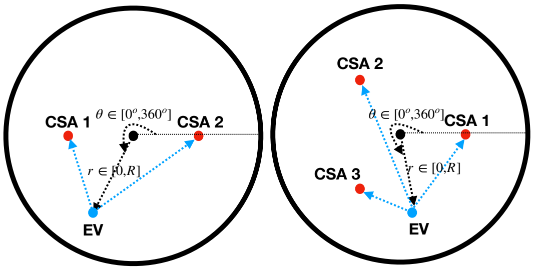


Figure 5. The travelling time sampling.

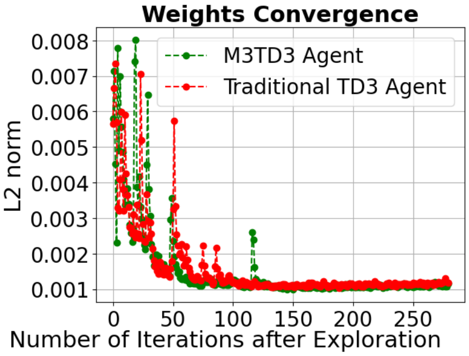
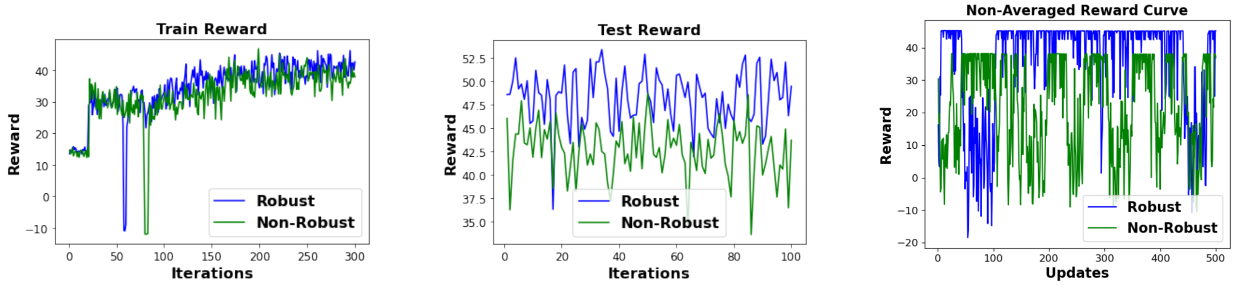


Figure 6. Convergence of Weights.

In Fig. 5, we illustrate the method used for sampling the travel time of each EV driver. For instance, if there are two EV CSAs within a circular area, each EV user's initial location is determined by uniformly sampling a radius between [0, 2km] and an angle from [0, 360]. Subsequently, the distances to the two CSAs are calculated. These distances are then divided by the average velocity to estimate the travel time. To estimate the waiting time $\hat{\tau}_w(t_a)$ and the price $\hat{\lambda}_t$, we use the 1-hidden layer NN regression model. To predict $\hat{\tau}_w(t_a)$

[1] https://electricgrids.engr.tamu.edu/activsg-time-series-data/



(a) Training Performance: Robust (M3TD3) vs Non-Robust (TD3).

(b) Testing Performance: Robust (M3TD3) vs Non-Robust (TD3).

(c) Non-averaged Testing Performance: Robust (M3TD3) vs Non-Robust (TD3).

Figure 7. M3TD3 Performance: 2-Agents.

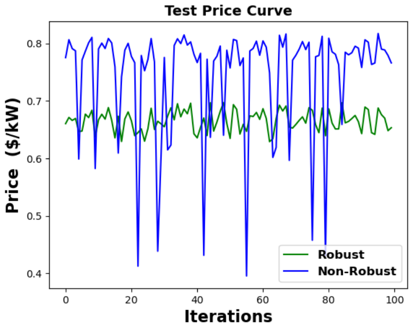


Figure 8. Charging prices λ given by two agents.

we use the waiting time $\hat{\tau}_w(t_o)$, count of EV drivers arriving at the station during $\tau_r(t_o)$, number of users departing during $\tau_r(t_o)$, and current charging prices at t_o as inputs. These factors help forecast the waiting time at t_a. We also use the previous 24-hour price samples to forecast the next 8-hour price samples, with the specified future price at t determined by the sum of estimated waiting time and travel time. The testing normalized mean square errors for waiting time and price are 0.0371 and 0.2377, respectively.

In practical scenarios, travel time holds greater significance for EV drivers compared to waiting time and pricing, as they tend to prioritize locating the nearest charging stations. Thus, we assign the weights w_p, w_r, and w_t as 2, 30, and 1, respectively. We also conduct an evaluation of the ground-truth EV station selection, which is based on the minimal score function, and compare it to the predicted EV station selection. The accuracy of the predicted EV station selection reaches 97.38% due to the overriding importance of travel time compared to the other two factors. We allocated 10 chargers and waiting seats in each station for the experiments below. We utilized Python 3.9.16 and PyTorch 1.11.0 to develop and train our CRL methods. All the algorithms were executed on a Mac OS machine equipped with an Arm-based M1 chip, 8 cores, and 16GB RAM for the development phase, and then using NVIDIA's RTX 2060 GPU.

EV charging requests follow a Poisson distribution with an average of 6 per hour, and charging periods uniformly span from 1 to 3 hours [2]. If a charging spot is vacant, the first waiting EV occupies it. New arrivals check charging availability first, move to waiting spots if all chargers are full, or else proceed to another station. This environment setup can generate a variety of scenarios ranging from both stations being fully occupied, one being occupied while the other is empty, or both having vacancies. Furthermore, every individual charging spot within the station can support a charging rate ranging from 4 kW to 10 kW.

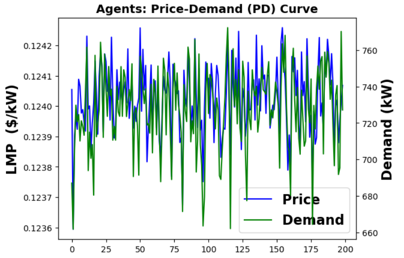
5.1. Hyperparameter Exploration

We conducted experiments based on the following general assumptions: the action space has a dimension of 11, which includes one price for newly arriving cars and ten charging rates for each of the currently occupied slots. Regarding command line parameters, the default values are configured as follows: the exploration noise is set to a standard normal Gaussian with a value of 0.1, the batch size is set to 256, the discount factor is 0.99, the policy noise (during critic update) is 0.2, the noise clip is 0.5, and a delayed policy (update) frequency of 2 is employed. After conducting numerous simulations to fine-tune the hyperparameters, we have selected a α value of 0.1, keeping $\alpha = 0$ for non-robust agents.

5.2. Two-Agent EV Setting

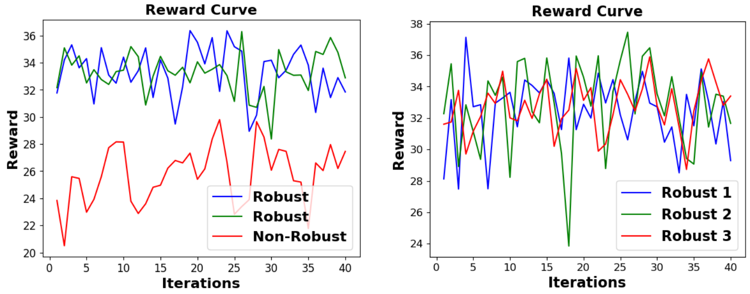
We utilized two agents to distinguish between the TD3 policy and the robust approach. Fig. 6 provides an initial insight into the convergence of our proposed method for both agents over the 300 iterations, 20 of which represent the pure exploration phase, while

[2] In this setting, it is almost impossible that all the charging stations are full.



(a) The total demand (including the base demands and charging demands) of the bus to which EV CSA Agent 1 is connected, along with its associated LMP, expressed in $/kW.

Figure 9. LMP-Demand Curves.



(a) Testing: 2 Robust (Green, Blue) vs Non-Robust (Red).

(b) Testing: 3 Robust Agents.

Figure 10. M3TD3 Performance: 3-Agents.

for the rest a mix of exploration and exploitation is employed. The L2 norm of the weights' difference approaches close to 0, with occasional spikes observed during the initial 70 iterations that gradually diminish over time. Throughout the training process depicted in Fig. 7(a), we computed the average performance across 500 updates. Interestingly, we observed no significant performance disparities between the two agents.

During the testing process, we observed a significant performance difference between the two agents, as illustrated in Fig. 7(b). The robust agent, represented in blue obtained an additional profit of over 20% compared to the non-robust one. Although at various points during the testing process, we observed brief periods when the traditional TD3 competitor obtained better profit averages, over the long term, the M3TD3 agent continued to average higher. A detailed look at Fig. 7(c)'s non-averaged graph reveals oscillating profit differences between the two agents. However, a performance average shows the robust M3TD3 agent outperforming its non-robust counterpart. Importantly, negative profits occur during periods with high marginal prices or significant demand prediction discrepancies.

In Fig. 8, the robust CSA not only provides prices that are more stable than those of the non-robust CSA, but also tends to be slightly cheaper most of the time, which is likely to draw a larger number of EV drivers to avail their services, thereby potentially increasing the CSA's profits. Figure 9 illustrates the correlation between power demand, encompassing both base, charging demands, and LMPs ($/kW). As depicted in Fig. 9, during periods of increased demand, power congestion is rising correspondingly. This relationship continues into Fig. 9, illustrating that higher levels of power congestion drive up the LMPs. The surge in LMPs is a mechanism to prevent power congestion, acting as a deterrent to overloading the system.

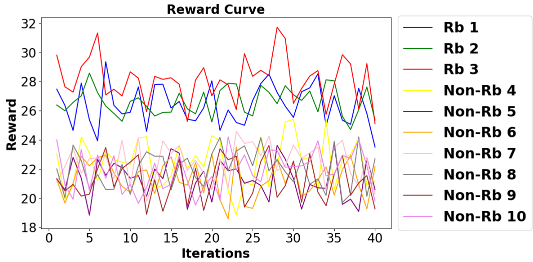


Figure 11. Testing Performance 10 Agents: Robust (M3TD3) vs Non-Robust (TD3).

5.3. K-Agent EV Setting

To better model the real-time electricity market, which typically involves a larger number of agents, we extended our analysis to include K-Agents. To demonstrate performance with noticeable differences among agents, we selected three agents for the analysis. Two of these agents employed the M3TD3 robust policy, while the third agent directly implemented the TD3 approach without considering robust actions.

For the three-agents cases, testing revealed superior overall performance of the robust agents (Green, Blue) compared to traditional TD3 policies, as illustrated in Fig. 10(a), and subsequently shown by average profits with $34.45 and $34.11 for the two M3TD3 agents versus $24.87 for the third, a 35% improvement. An additional plot represents the environment populated by three robust agents, each trying to maximize their minimal rewards. The training process has a similar curve with the scenarios above, while Fig. 10(b) illustrates the testing scenario. We observe that the curves fit much closer to each other, with profits being spread more even between the agents. In Fig. 11, the scalability of the proposed method is demonstrated with 10 agents, of which 3 are robust and the remaining 7 are non-robust. In terms of computational cost, the runtime scales linearly with respect to the number of agents, spending 360 minutes to solve a 100-agent instance with majority of time spent solving the OPF problem.

6. Conclusion

In this paper, we adapted a robust MARL approach, M3TD3 to efficiently solve the Charging Station Dynamic Pricing and Scheduling (CSDPS) problem in a highly dynamic and uncertain EV charging market. Our approach, which employs minimax policy gradient in competitive reinforcement learning settings, yielded up to 35% additional profit compared to non-robust RL managed stations that only maximize expected profits. By coupling our problem with the distribution network using LMPs, we ensure efficient and reliable grid operation while promoting the use of renewable energy resources, resulting in reduced costs for both the DSO and EV charging station management companies.

References

[1] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, 2019. DOI: 10.1109/TSG.2018.2879572.

[2] M. Yilmaz and P. T. Krein, "Review of benefits and challenges of vehicle-to-grid technology," in *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, IEEE, 2012, pp. 3082–3089.

[3] L. Bird, M. Milligan, and D. Lew, "Integrating variable renewable energy: Challenges and solutions," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2013.

[4] P. Siano, "Demand response and smart grids—a survey," *Renewable and sustainable energy reviews*, vol. 30, pp. 461–478, 2014.

[5] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 152–178, 2014.

[6] M. Tursini, F. Parasiliti, G. Fabri, and E. Della Loggia, "A fault tolerant e-motor drive system for auxiliary services in hybrid electric light commercial vehicle," in *2014 IEEE International Electric Vehicle Conference (IEVC)*, IEEE, 2014, pp. 1–6.

[7] W. Tang, S. Bi, and Y. J. Zhang, "Online coordinated charging decision algorithm for electric vehicles without future information," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2810–2824, 2014.

[8] N. Zou, L. Qian, and H. Li, "Auxiliary frequency and voltage regulation in microgrid via intelligent electric vehicle charging," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, 2014, pp. 662–667.

[9] N. Liu, Q. Chen, J. Liu, *et al.*, "A heuristic operation strategy for commercial building microgrids containing evs and pv system," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2560–2570, 2014.

[10] *Charging network rankings in us.* [Online]. Available: https://evadoption.com/ev-charging-stations-statistics/us-charging-network-rankings/.

[11] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, PMLR, 2018, pp. 1587–1596.

[12] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 4213–4220.

[13] P. J. Hoen, K. Tuyls, L. Panait, S. Luke, and J. A. La Poutre, "An overview of cooperative and competitive multiagent learning," in *Learning and Adaption in Multi-Agent Systems: First International Workshop, LAMAS 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers*, Springer, 2006, pp. 1–46.

[14] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," *arXiv preprint arXiv:2011.00583*, 2020.

[15] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[16] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[17] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*, Elsevier, 1994, pp. 157–163.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.