Mobility-Aware Charging Scheduling for Shared On-Demand Electric Vehicle Fleet Using Deep Reinforcement Learning

Yanchang Liang, *Student Member, IEEE*, Zhaohao Ding, *Senior Member, IEEE*, Tao Ding, *Senior Member, IEEE*, and Wei-Jen Lee, *Fellow, IEEE*

***Abstract*—With the emerging concept of sharing-economy, shared electric vehicles (EVs) are playing a more and more important role in future mobility-on-demand traffic system. This article considers joint charging scheduling, order dispatching, and vehicle rebalancing for large-scale shared EV fleet operator. To maximize the welfare of fleet operator, we model the joint decision making as a partially observable Markov decision process (POMDP) and apply deep reinforcement learning (DRL) combined with binary linear programming (BLP) to develop a near-optimal solution. The neural network is used to evaluate the state value of EVs at different times, locations, and states of charge. Based on the state value, dynamic electricity prices and order information, the online scheduling is modeled as a BLP problem where the decision variables representing whether an EV will 1) take an order, 2) rebalance to a position, or 3) charge. We also propose a constrained rebalancing method to improve the exploration efficiency of training. Moreover, we provide a tabular method with proved convergence as a fallback option to demonstrate the near-optimal characteristics of the proposed approach. Simulation experiments with real-world data from Haikou City verify the effectiveness of the proposed method.**

***Index Terms*—Electric vehicle, deep reinforcement learning, order dispatching, rebalancing, charging scheduling.**

NOMENCLATURE

Acronyms

BLP	Binary linear programming
CR	Constrained rebalancing
DRL	Deep reinforcement learning
EV	Electric vehicle
MDP	Markov decision process
POMDP	Partially observable Markov decision process
REV	Revenue-based method
RL	Reinforcement learning
SOC	State of charge
TD	Temporal-difference.

Sets and Indices

$\mathcal{A}_{g,t}$	Action space in grid g at time step t
$\mathcal{G}$	Set of grids $\{1, 2, \ldots, G\}$ indexed by g, h
$\mathcal{G}^{\text{ch}}$	Set of grids equipped with chargers, $\mathcal{G}^{\text{ch}} \subseteq \mathcal{G}$
$\mathcal{I}_{g,t}$	Set of available EVs $\{1, 2, \ldots, I_{g,t}\}$ indexed by j in grid g at time step t
$\mathcal{J}_{g,t}$	Set of available dispatches $\{1, 2, \ldots, J_{g,t}\}$ indexed by j in grid g at time step t
$\mathcal{J}^{\text{ch}}_{g,t}$	Set of charging dispatches, $\mathcal{J}^{\text{ch}}_{g,t} \subseteq \mathcal{J}_{g,t}$
$\mathcal{J}^{\text{or}}_{g,t}$	Set of order dispatches, $\mathcal{J}^{\text{or}}_{g,t} \subseteq \mathcal{J}_{g,t}$
$\mathcal{J}^{\text{re}}_{g,t}$	Set of rebalancing dispatches, $\mathcal{J}^{\text{re}}_{g,t} \subseteq \mathcal{J}_{g,t}$
t	Index for time steps.

Parameters

$I^{\text{re}}_{g\to h,t}$	Number of EVs rebalanced from g to h at time step t
$I^{\text{in-ch}}_{g,t}$	Number of EVs being charged in g at time step t
$J^{\text{or}}_{g,t}$	Number of order dispatches in g at time step t
l_j	Destination of dispatch j
N^{ch}_g	Total number of chargers in grid g
$p^{\text{ch}}_{g,t}$	Charging price of grid g at time step t
p^{or}_j	Price of order dispatch j.

Variables

$\boldsymbol{a}_{g,t}$	Joint action of all available EVs in g at time step t
$\boldsymbol{s}_{g,t}$	Joint state of all available EVs in g at time step t
ΔE_{ij}	Change in SOC during EV i execution dispatch j
Δt_{ij}	Duration for EV i to complete dispatch j
$a_{i,t}$	Action of EV i at time step t
b_{ij}	Binary decision variable ($b_{ij} = 1$ if EV i matches dispatch j, else 0)
$E_{i,t}$	SOC of EV i at time step t
$l_{i,t}$	Location of EV i at time step t
$R_{i,t}$	Return of EV i following time step t
$r_{i,t}$	Reward of EV i at time step t

Manuscript received April 10, 2020; revised August 1, 2020; accepted September 9, 2020. Date of publication September 21, 2020; date of current version February 26, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 51907063; in part by the Fundamental Research Funds for the Central Universities under Grant 2019MS054; and in part by the Support Program for the Excellent Talents in Beijing City under Grant X19048. Paper no. TSG-00531-2020. *(Corresponding author: Zhaohao Ding.)*

Yanchang Liang and Zhaohao Ding are with the School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China (e-mail: liangyancang@gmail.com; zhaohao.ding@ncepu.edu.cn).

Tao Ding is with the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: tding15@xjtu.edu.cn).

Wei-Jen Lee is with the Energy Systems Research Center, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: wlee@uta.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2020.3025082

r_{ij}	Cumulative discounted reward during EV i execution dispatch j
$s_{i,t}$	State of EV i at time step t
s_{ij}	State of EV i after completing dispatch j.

Functions

$\pi(\boldsymbol{s})$	Deterministic policy function
$V(s;\theta)$	State value function approximated by the neural network with parameters θ.
$V_{\pi}^{i}(s)$	State value function of EV i under policy π.

I. Introduction

In RECENT years, the advent of large-scale shared on-demand ride-hailing platforms such as Uber and Didi Chuxing has greatly transformed the way people travel. At the same time, electric vehicles (EVs) have received widespread attention due to their lower pollution emissions and lower energy costs compared to gasoline vehicles, but the limited range and availability of charging infrastructure has hindered the application of EVs in private mobility. However, in combination with shared mobility-on-demand technology, managing EVs in the form of a fleet can ensure that there are sufficient number of vehicles with adequate energy to satisfy customers' travel demands through intelligent charging scheduling, order dispatching and vehicle rebalancing, thereby eliminating "range anxiety". Moreover, the development of such shared mobility-on-demand fleets in large cities also has positive effects on mitigating parking and traffic congestion problems [1].

In this article, we consider the welfare maximization problem for a shared on-demand EV fleet operator. In general, there are three main scheduling tasks for operating an EV fleet, namely (i) order dispatching, i.e., to match the orders and vehicles in real time to directly deliver the service to the users, (ii) vehicle rebalancing, to reposition some idle vehicles to other locations, and (iii) charging scheduling, i.e., to determine the charging location and time for each EV. These three types of scheduling are closely related to each other, e.g., EVs can choose appropriate order dispatching or rebalancing to transfer to a location with greater demand to increase future revenue, or to a charging station with low electricity prices to reduce charging costs. Some recent work has jointly optimized different types of scheduling. Zhang *et al.* [2] proposed a model predictive control (MPC) approach that optimizes order dispatching and rebalancing of the fleet subject to energy constraints. Duan *et al.* [3] optimized order dispatching and rebalancing based on network flow model. In addition, Tsao *et al.* [4] proposed a stochastic MPC algorithm focusing on vehicle rebalancing. These methods have shown promising results. However, the detailed transportation network model may result in scalability issue.

Recent years witnessed tremendous success in reinforcement learning (RL) in modeling computational challenging decision-making problems [5]–[7] that were previously intractable. The computational overhead required by RL to solve the multi-agent resource management problem is usually much smaller than that of operations research methods, which makes RL more suitable for real-time scheduling of large-scale fleet [8]. As a model-free method, the training process of RL is based on historical experience (e.g., historical order data [9]) without the need to model transportation networks. In addition, RL is a commonly used method for sequential decision-making problems, which can well balance the fleet's immediate and future revenue. For example, greedily matching vehicles with long-distance orders can receive high immediate gain at a single order dispatching stage, but it might jeopardize future revenue because of its long drive time and unpopular destination [10]. However, one of the basic concepts of RL—state value—can naturally evaluate the future revenue of the vehicle in a certain spatiotemporal state.

Recently, RL has been applied to order dispatch of ride-hailing platforms [8], [10]–[13]. In [8], a learning and planning method was proposed and successfully deployed in Didi Chuxing's production system, where the offline learning step applies dynamic programming to update the state value stored in a table, and the online planning step uses the state value to compute real-time matching through the Kuhn-Munkres (KM) algorithm [14]. Since tabular methods are severely limited by the curse of dimensionality, Shi *et al.* [11] used neural networks to replace the tabular method in [8] to represent state value function. For the same purpose, Tang *et al.* [12] proposed a new neural network structure, Cerebellum Value Network, to represent the state value function. RL combined with neural networks is often called deep RL (DRL). To capture dynamic demand-supply variations, mean field DRL [15] was used to approximate the interaction between the vehicle and its surrounding area [13].

RL was also applied to fleet rebalancing [16] and charging scheduling [17], [18]. Lin *et al.* [16] proposed a contextual multi-agent DRL framework for rebalancing large-scale fleet. Vandael *et al.* [17] used RL to solve the charging scheduling problem of EV fleet. Ding *et al.* [18] proposed a DRL-based EV charging strategy to maximize the profit of the distribution system operators while satisfying all the physical constraints. In addition, Wan *et al.* [19] applied DRL to optimize charging/discharging scheduling for private EV.

Few works [10], [11], [20] used RL method to jointly optimize different types of scheduling for EV fleet. Jin *et al.* [10] optimized joint order dispatching and rebalancing, but due to the limited stability of the hierarchical DRL used, its application in the real world is very challenging [10]. In addition, they did not consider charging scheduling. Shi *et al.* [11] considered charging scheduling while optimizing order dispatching, but they used a positive constant reward to guide EV charging, which could not reflect the impact of locational electricity prices on charging behavior. The adopted KM algorithm presents challenges on incorporating constraints for charging behavior (e.g., the number of available chargers should be limited). In addition, they did not consider the rebalancing of EV fleet. Turan *et al.* [20] used DRL to solve the joint routing, charging and pricing problem of EV fleet, where the input of neural network is the state of entire study area, and the output is the scheduling result of all EVs. But the state and action space of such a completely centralized scheduling would be immensely huge. For instance,

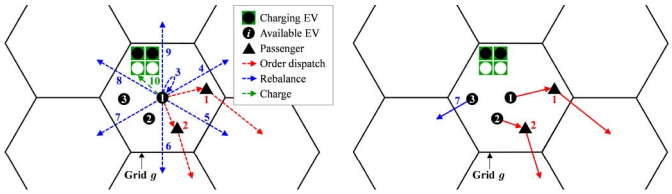
when the study area is divided into 10 nodes and the state of charge (SOC) is discrete to 6 levels, the dimension of state space has reached 1240 [20].

In order to facilitate joint optimization of different types of scheduling decisions, we treat each rebalancing or charging behavior as an order and assign it to EVs in the form of dispatch. The difference is that one order dispatch is assigned to at most one EV, but one rebalancing or charging dispatch can be assigned to multiple EVs. For large-scale EV fleet, the state and decision space of the joint optimization problem is huge and dynamically changing, and cannot be solved directly using existing DRL algorithms (e.g., DQN [5], DDPG [21], SAC [22]). To this end, we model the joint optimization problem as a partially observable Markov decision process (POMDP) [23], in which we evaluate the state value of each EV rather than the entire area to reduce the state space (i.e., *policy evaluation*). In order to reduce the decision space, we divide the study area into hexagonal grids for decentralized scheduling. We argue that this scheduling problem is different from the conventional Markov decision process (MDP) in that the state transition can be determined in advance. With this feature, we transform the collaborative scheduling of EVs in each grid into a binary linear programming (BLP) problem (i.e., *policy improvement*). We interactively perform policy evaluation and policy improvement (a common form of RL) [24] to develop a near-optimal scheduling policy. In policy evaluation, we use techniques such as neural network approximation and empirical replay mechanisms to improve computational and data efficiency [5], but lose the theoretical guarantee of convergence. We thus provide a convergent tabular method as a fallback option. An experimental comparison with the tabular method demonstrates that the proposed method has achieved near-optimal performance.

Our major contributions are listed as follows:

1) We propose a joint optimization framework for a large-scale shared on-demand EV fleet operator as a POMDP which simultaneously considers charging scheduling, order dispatching and vehicle rebalancing decisions.
2) We develop a solution method utilizing DRL combined with BLP to obtain a near-optimal scheduling policy. The proposed method is tested with city-scale real-world data to illustrate its effectiveness. We also provide a convergent tabular method as a fallback option and give a proof of its convergence.
3) We propose a constraint method for vehicle rebalancing to reduce the exploration space in DRL training process, which can also be used to improve the performance of conventional scheduling methods.

The rest of this article is organized as follows. We formulate the joint optimization problem as a POMDP in Section II. Section III describes the specific implementation methods of policy improvement and policy evaluation. In Section IV, simulation experiments based on real-world data from Haikou City are used to verify the effectiveness of the proposed method. Finally, we conclude the paper in Section V.



(a) (b)

Fig. 1. Illustration of joint order dispatching, rebalancing, and charging scheduling in grid g at time step t. (a) All available dispatches; (b) Dispatches assigned to EVs.

II. A POMDP Formulation

This article considers joint order dispatching, vehicle rebalancing, and charging scheduling for shared on-demand EV fleet operator. As shown in Fig. 1, the study area is divided into hexagonal grids, denoted as $\mathcal{G} = \{1, 2, \ldots, G\}$, and each grid can serve as a trip origin or destination. The hexagonal grid has been widely used for fleet scheduling problems [10], [12], [16] and its advantages are discussed in [25], [26]. We perform scheduling for EV fleet at a series of discrete time steps, $t = 0, 1, \ldots, T-1$ (T is the termination time of episode). The objective of scheduling optimization is to maximize the operator's profit, i.e., to increase the total order revenue and reduce the total charging cost. We consider each EV as an agent and model the joint scheduling problem as a POMDP in a fully cooperative setting, where the major components are defined as follows.

1) *State:* We maintain a set of available EVs (not in service or charging) for each grid $g \in \mathcal{G}$, denoted as $\mathcal{I}_{g,t} = \{1, 2, \ldots, I_{g,t}\}$. For example, the available EVs set for grid g in Fig. 1 is $\mathcal{I}_{g,t} = \{1, 2, 3\}$. Since EVs are moving and entering/exiting available status at any time, set $\mathcal{I}_{g,t}$ changes over time. For each available EV $i \in \mathcal{I}_{g,t}$, its state variable $s_{i,t}$ consists of the current time step t, its location $l_{i,t}$ and SOC $E_{i,t}$, i.e., $s_{i,t} = [t, l_{i,t}, E_{i,t}]$. The location $l_{i,t}$ is represented by the index of the grid where EV i is located, i.e., $l_{i,t} = g \ \forall i \in \mathcal{I}_{g,t}$. Note that when an EV is not available, we ignore its state. In addition, we use the state vector $\boldsymbol{s}_{g,t} = [s_{i,t}]_{i \in \mathcal{I}_{g,t}}$ to represent the states of all available EVs in grid g.

2) *Action:* To facilitate joint optimization, we model each rebalancing or charging decision as an order and assign it to EVs in the form of dispatch. Specifically, we maintain a time-varying available dispatch set $\mathcal{J}_{g,t} = \{1, 2, \ldots, J_{g,t}\}$ for each grid g, which includes order set $\mathcal{J}^{\text{or}}_{g,t}$, rebalancing set $\mathcal{J}^{\text{re}}_{g,t}$, and charging set $\mathcal{J}^{\text{ch}}_{g,t}$, i.e., $\mathcal{J}_{g,t} = \mathcal{J}^{\text{or}}_{g,t} \cup \mathcal{J}^{\text{re}}_{g,t} \cup \mathcal{J}^{\text{ch}}_{g,t}$. For example, the available dispatch set for grid g in Fig. 1 includes $\mathcal{J}^{\text{or}}_{g,t} = \{1, 2\}$, $\mathcal{J}^{\text{re}}_{g,t} = \{3, 4, 5, 6, 7, 8, 9\}$, and $\mathcal{J}^{\text{ch}}_{g,t} = \{10\}$. At each time step t, a dispatch decision from set $\mathcal{J}_{g,t}$ will be assigned to EV i, and this action is represented as a vector $a_{i,t} = [b_{ij}]_{j \in \mathcal{J}_{g,t}}$, where element b_{ij} is a binary variable ($b_{ij} = 1$ if EV i matches dispatch j, else 0). For example, in Fig. 1(b), the action of EV 3 matching dispatch 7 is denoted as $a_{3,t}$ = [0 0 0 0 0 0 1 0 0 0].

We utilize matrix $\boldsymbol{a}_{g,t} = [b_{ij}]_{i \in \mathcal{I}_{g,t}, j \in \mathcal{J}_{g,t}}$ to represent the joint action of all available EVs in grid g. For the example shown

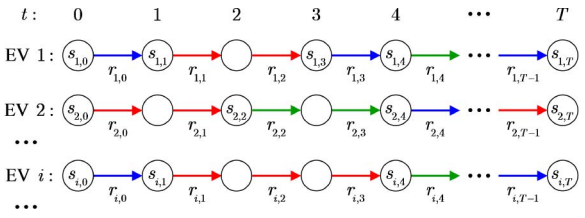


Fig. 2. State transition process of each EV.

in Fig. 1(b),

$$\boldsymbol{a}_{g,t} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \\ a_{3,t} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

The generation of joint action needs to meet certain dispatch constraints, which we will discuss in Section III-A. We denote the action space that satisfies the dispatch constraints as $\mathcal{A}_{g,t}$. Note that unlike the standard MDP, the action space $\mathcal{A}_{g,t}$ changes over time as it is affected by available EVs and available dispatches.

3) *Reward and Return:* Reward $r_{i,t}$ is defined as the revenue (cost is treated as negative revenue) obtained by EV i in time step t. Return $R_{i,t}$ is defined as the cumulative discounted rewards of EV i from time step t until the end of the episode:

$$R_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \cdots + \gamma^{T-t-1} r_{i,T-1} \quad (2)$$

where $\gamma \in [0, 1]$ is the discount factor that is used to trade off the importance between immediate and future rewards.

As shown in Fig. 2, the number of time steps EV i takes to complete dispatch j, denoted as Δt_{ij}, may sometimes be greater than 1, i.e., EV i requires multiple time steps to reach the next state, which is also called semi-MDP [27]. Similar to the standard MDP, there is a recursive relationship between returns in adjacent states:

$$\begin{aligned} R_{i,t} &= r_{i,t} + \gamma r_{i,t+1} + \cdots + \gamma^{\Delta t_{ij}-1} r_{i,t+\Delta t_{ij}-1} \\ &\quad + \gamma^{\Delta t_{ij}} \left(r_{i,t+\Delta t_{ij}} + \cdots + \gamma^{T-t-\Delta t_{ij}-1} r_{i,T-1} \right) \\ &= r_{ij} + \gamma^{\Delta t_{ij}} R_{i,t+\Delta t_{ij}} \end{aligned} \quad (3)$$

where r_{ij} is the cumulative discounted reward during EV i execution dispatch j. For example, if EV i matches the dispatch j that starts at $t = 1$ and has a duration $\Delta t_{ij} = 3$ (as shown in Fig. 2), then $r_{ij} = r_{i,1} + \gamma r_{i,2} + \gamma^2 r_{i,3}$, and $R_{i,1} = r_{ij} + \gamma^3 R_{i,4}$.

4) *State Transition* In the fleet operation platform, if EV i matches dispatch j, the relevant information—dispatch destination l_j, dispatch duration Δt_{ij}, change in SOC ΔE_{ij}, and reward r_{ij}—can be calculated before the dispatch is performed:

- $j \in \mathcal{J}^{\text{or}}_{g,t}$: The destination l_j, duration Δt_{ij}, price p^{or}_j, and mileage of order dispatch j can be obtained by the fleet operation platform. The change in SOC ΔE_{ij} can be calculated from the mileage ($\Delta E_{ij} < 0$). For an order with a price of p^{or}_j and a duration of Δt_{ij}, the reward equally allocated to each time step is $p^{\text{or}}_j / \Delta t_{ij}$, and the cumulative discounted reward for this order is

$$r_{ij} = \sum_{\tau=t}^{t+\Delta t_{ij}-1} \gamma^{\tau-t} \frac{p^{\text{or}}_j}{\Delta t_{ij}} \quad (4)$$

- $j \in \mathcal{J}^{\text{re}}_{g,t}$: In this article, staying in the current hexagonal grid is modeled as a special kind of rebalancing. Therefore, the destination l_j of rebalancing dispatch j is the current grid or an adjacent grid. The time steps required to complete the rebalancing dispatch (Δt_{ij}) can be calculated in real time based on traffic conditions. For simplicity, in this paper we set $\Delta t_{ij} = 1$ [10], [16]. The change in SOC ΔE_{ij} can also be calculated from the mileage ($\Delta E_{ij} \leq 0$). Since rebalancing is not gainable or payable, we set $r_{ij} = 0$.
- $j \in \mathcal{J}^{\text{ch}}_{g,t}$: The charging dispatch destination l_j is still the current grid g. It is assumed that once each EV i is dispatched to charge, it will not leave until it is fully charged. Therefore, the change in SOC $\Delta E_{ij} = 1 - E_{i,t}$, and the charging duration $\Delta t_{ij} = \lceil \Delta E_{ij} / P_{\text{ch}} \rceil$. The charging cost per time step is the product of the electricity price and the change in SOC. The cumulative discounted reward during charging is the inverse of the total charging cost:

$$r_{ij} = - \sum_{\tau=t}^{t+\Delta t_{ij}-1} \gamma^{\tau-t} p^{\text{ch}}_{g,\tau} \Delta E_{i,\tau} \quad (5)$$

where $p^{\text{ch}}_{g,\tau}$ is the dynamic charging price of grid g in time step τ, which is assumed to be obtainable or predictable. $\Delta E_{i,\tau}$ is the change in SOC of EV i in time step τ, which is equal to the charging power P_{ch} if SOC $E_{i,\tau}$ is not greater than 1, i.e., $\Delta E_{i,\tau} = \min(P_{\text{ch}}, 1 - E_{i,\tau})$.

In short, each EV i whose state is $s_{i,t} = [t, l_{i,t}, E_{i,t}]$ at time step t will transition to state $s_{i,t+\Delta t_{ij}} = [t + \Delta t_{ij}, l_j, E_{i,t} + \Delta E_{ij}]$ (abbreviated as s_{ij}) at time step $t + \Delta t_{ij}$ after completing dispatch j, and obtain a cumulative discounted reward of r_{ij} during this period. Both the reward r_{ij} and the new state s_{ij} can be determined before dispatch j is executed.

5) *Policy and State Value Function:* From the perspective of EV fleet operator, the goal of the joint optimization problem should be to maximize the expected return of the entire fleet. However, the state and action space of scheduling all EVs completely centrally is too large [20]. A common solution is to use a partially observable setting that ignores the states of other EVs when scheduling for each EV [8], [11]. Considering that there are mutual constraints on the actions of EVs in a grid, e.g., the number of available orders and chargers is limited, we jointly schedule EVs in each grid to facilitate their cooperation. The scheduling objective for each grid can be formulated as

$$\max_{\pi} \sum_{i \in \mathcal{I}_{g,t}} \mathbb{E}_{\pi} \left[R_{i,t} | s_{i,t} \right] = \max_{\pi} \sum_{i \in \mathcal{I}_{g,t}} V^{i}_{\pi} \left(s_{i,t} \right) \quad (6)$$

where π is a deterministic policy that maps from the state of each grid to a joint scheduling action, i.e., $\boldsymbol{a}_{g,t} = \pi(\boldsymbol{s}_{g,t})\ \forall g \in \mathcal{G}$. The expected return when EV i starts in $s_{i,t}$ and follows π thereafter, i.e., $\mathbb{E}_{\pi}[R_{i,t}|s_{i,t}]$, is defined as the value function of state $s_{i,t}$, denoted as $V^{i}_{\pi}(s_{i,t})$. According to (3), the relationship between the value functions of adjacent states (i.e., Bellman equation for semi-MDP [27]) is

$$\begin{aligned} V^{i}_{\pi} \left(s_{i,t} \right) &= \mathbb{E}_{\pi} \left[R_{i,t} | s_{i,t} \right] = \mathbb{E}_{\pi} \left[r_{ij} + \gamma^{\Delta t_{ij}} R_{i,t+\Delta t_{ij}} | s_{i,t} \right] \\ &= \mathbb{E}_{\pi} \left[r_{ij} + \gamma^{\Delta t_{ij}} V^{i}_{\pi} \left(s_{i,t+\Delta t_{ij}} \right) | s_{i,t} \right]. \end{aligned} \quad (7)$$

III. Scheduling Policy

Policy evaluation and policy improvement are performed interactively to develop a near-optimal scheduling policy. Policy evaluation refers to the computation of the value functions for a given policy. Policy improvement refers to the computation of an improved policy given the value function for that policy. We will elaborate the implementation of these two processes in the proposed scheduling problem. For simplicity, we omit the subscript t for all variables in this section.

A. Policy Improvement With Binary Linear Programming

Policy improvement is done by making the policy greedy with respect to the current value function [24]. We can formulate an improved greedy policy based on the sum of the state values of all available EVs in a grid:

$$\pi'(\boldsymbol{s}_g) = \arg\max_{\boldsymbol{a}_g \in \mathcal{A}_g} \sum_{i \in \mathcal{I}_g} \mathbb{E}\left[r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij}) | s_i, a_i \right] \quad (8a)$$

$$= \arg\max_{\boldsymbol{a}_g \in \mathcal{A}_g} \sum_{i \in \mathcal{I}_g} \left[r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij}) \right] \quad (8b)$$

The reason why (8a) is equal to (8b) is that the reward r_{ij} and the new state s_{ij} obtained after taking the action a_i are determinable in the current scheduling problem (Section II-D). In (8b), r_{ij} is the instant reward of EV i execution dispatch j, and $V_\pi^i(s_{ij})$ reflects the future rewards of EV i after executing dispatch j to reach a new state. Therefore, Eq. (8b) aims to take the action that maximizes the sum of short-term and long-term rewards of all EVs. The reward of EVs after episode ends is not considered, so when s_{ij} is the termination state of episode (corresponding time step $t + \Delta t_{ij} \geq T$), $V_\pi^i(s_{ij}) = 0$. Therefore, we rewrite (8b) as

$$\pi'(\boldsymbol{s}_g) = \arg\max_{\boldsymbol{a}_g \in \mathcal{A}_g} \sum_{i \in \mathcal{I}_g} y_{ij} \quad (9)$$

where

$$y_{ij} = \begin{cases} r_{ij}, & s_{ij} \text{ is terminal} \\ r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij}), & s_{ij} \text{ is non-terminal} \end{cases} \quad (10)$$

We use dispatch constraints to represent action space $\mathcal{A}_g$, and use binary variable b_{ij} to directly represent the choice of dispatch. Then Eq. (9) can be rewritten as a BLP problem with decision variable $\boldsymbol{a}_g = [b_{ij}]_{i \in \mathcal{I}_g, j \in \mathcal{J}_g}$:

$$\arg\max_{b_{ij}} \sum_{i \in \mathcal{I}_g} \sum_{j \in \mathcal{J}_g} y_{ij} b_{ij} \quad (11)$$

subject to:

$$b_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I}_g, \ \forall j \in \mathcal{J}_g \quad (11a)$$

$$\sum_{j \in \mathcal{J}_g} b_{ij} = 1 \quad \forall i \in \mathcal{I}_g \quad (11b)$$

$$\sum_{i \in \mathcal{I}_g} b_{ij} \leq 1 \quad \forall j \in \mathcal{J}_g^{\text{or}} \quad (11c)$$

$$\sum_{i \in \mathcal{I}_g} \sum_{j \in \mathcal{J}_g^{\text{ch}}} b_{ij} \leq N_g^{\text{ch}} - I_g^{\text{in-ch}} \quad (11d)$$

$$b_{ij}\left(E_i + \Delta E_{ij} + \Delta E_{l_j}^{\mapsto \text{ch}} \right) \geq 0 \ \forall i \in \mathcal{I}_g, \forall j \in \mathcal{J}_g^{\text{or}} \cup \mathcal{J}_g^{\text{re}} \quad (11e)$$

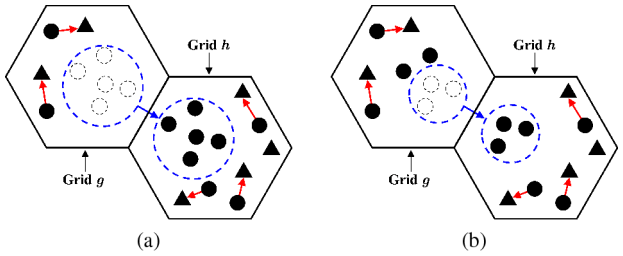


Fig. 3. EVs rebalance from grid g to h. (a) Unconstrained rebalancing; (b) Constrained rebalancing.

$$\sum_{i \in \mathcal{I}_g} \sum_{\substack{j \in \mathcal{J}_g^{\text{re}} \\ l_j \neq g}} b_{ij} \leq \max\left(0, \left\lceil \frac{\left(J_{l_j}^{\text{or}} - I_{l_j} \right) - \left(J_g^{\text{or}} - I_g \right)}{2} \right\rceil \right) \quad (11f)$$

where Eq. (11a) constrains each b_{ij} to a binary variable: If dispatch j is assigned to EV i, then $b_{ij} = 1$, otherwise $b_{ij} = 0$. Eq. (11b) indicates that each EV must be assigned with one dispatch. Eq. (11c) indicates that each order dispatch is assigned to at most one EV. In (11d), N_g^{ch} is the total number of chargers in grid g ($N_g^{\text{ch}} \geq 0$), and $I_g^{\text{in-ch}}$ is the number of EVs being charged in grid g. This means that the number of EVs dispatched to charge in grid g cannot be greater than the number of available chargers. In (11e), $\Delta E_{l_j}^{\mapsto \text{ch}}$ represents the change in SOC during the journey from grid l_j to the nearest charger ($\Delta E_{l_j}^{\mapsto \text{ch}} \leq 0$), which ensures that each EV has a sufficient charge level to reach the nearest charger after completing an order or rebalancing. Eq. (11f) is a constraint on the number of rebalanced EVs, which is used to reduce the action space to improve the efficiency of exploration in policy evaluation, which will be described in detail next.

For the two grids g and h as shown in Fig. 3, the destination of rebalancing dispatch in grid g is the original grid g or the adjacent grid h. If grid h has more incoming orders than g for a period of time in the future, rebalancing EV i to h is more likely to obtain higher return than g (as the energy cost due to moving to an adjacent grid is relatively small compared with service revenue), i.e., $V_\pi^i(s_{ij})|_{l_j=h} > V_\pi^i(s_{ij})|_{l_j=g} \ \forall i \in \mathcal{I}_g$. In addition, the rewards for all rebalancing dispatches are 0, i.e., $r_{ij}|_{l_j=h} = r_{ij}|_{l_j=g} = 0$. Therefore, according to (10) and (11), each EV (except assigned to an order or a charging dispatch) will be rebalanced to grid h to increase the objective function, which will make the order demand in grid g unsatisfactory (as shown in Fig. 3(a)). In addition, during the evaluation of value function, some errors, such as $V_\pi^i(s_{ij})|_{l_j=h}$ is sometimes larger than, and sometimes smaller than $V_\pi^i(s_{ij})|_{l_j=g}$, will make some EVs repeatedly rebalance in the two grids, which results in a waste of energy.

To cope with this problem, we constrain the number of rebalanced EVs by balancing the demand-supply gap between adjacent grids. At time step t, the number of available EVs in grid g, h is denoted as I_g, I_h, and the number of order dispatches is denoted as $J_g^{\text{or}}, J_h^{\text{or}}$, respectively. Then the demand-supply gap in these two grids is $J_g^{\text{or}} - I_g$ and $J_h^{\text{or}} - I_h$, respectively. Without any rebalancing between the two grids, we predict that the demand-supply gap in the two grids at time

step $t+1$ will still be $J_g^{\text{or}} - I_g$ and $J_h^{\text{or}} - I_h$. In order to satisfy more orders as a whole, we assume that after rebalancing $I_{g\to h}^{\text{re}}$ EVs from g to h at time step t, the demand-supply gap between the two grids will be balanced at time step $t+1$ (as shown in Fig. 3(b)):

$$J_g^{\text{or}} - \left(I_g - I_{g\to h}^{\text{re}}\right) = J_h^{\text{or}} - \left(I_h + I_{g\to h}^{\text{re}}\right) \tag{12}$$

Then we have

$$I_{g\to h}^{\text{re}} = \frac{\left(J_h^{\text{or}} - I_h\right) - \left(J_g^{\text{or}} - I_g\right)}{2} \tag{13}$$

Note that when the demand-supply gap of grids g and h are both positive, $I_{g\to h}^{\text{re}}$ may still be a positive number, but at this time, the best policy in grid g may be to let all EVs serve the orders, i.e., $I_{g\to h}^{\text{re}}$ should be 0. Therefore, we do not consider $I_{g\to h}^{\text{re}}$ obtained by (13) as the specific number of rebalanced EVs but the upper limit number. Since $I_{g\to h}^{\text{re}}$ may also be a fraction or a negative number, we round it up and set it to not less than 0, and finally write it as the constraint form shown in (11f). Adding reasonable constraints to rebalancing in this way can eliminate unreasonable actions in the action space to improve the efficiency of exploration, which we call constrained rebalancing (CR). Note that CR can also be applied to conventional scheduling policies.

B. Policy Evaluation With Neural Networks

Given an estimated value function $\tilde{V}(s)$ for policy π, the process of policy evaluation is to change it to be more like the true value function $V_\pi(s)$ for policy π. Temporal-difference (TD) prediction [28] is a commonly used method for policy evaluation:

$$\tilde{V}(s_i) \leftarrow \tilde{V}(s_i) + \alpha\left[r_{ij} + \gamma^{\Delta t_{ij}}\tilde{V}\left(s_{ij}\right) - \tilde{V}(s_i)\right] \tag{14}$$

where α is the step size. $r_{ij} + \gamma^{\Delta t_{ij}}\tilde{V}(s_{ij}) - \tilde{V}(s_i)$, called the *TD error*, is used to measure the difference between the estimated value of s_i and the better estimated $r_{ij} + \Delta t_{ij}\tilde{V}(s_{ij})$. $r_{ij} + \Delta t_{ij}\tilde{V}(s_{ij})$ is also called *TD target value*. TD prediction bases its update in part on an existing estimate, so it is a *bootstrapping* method [24].

Generally speaking, the value function can be a look-up table that directly stores values of states. However, in the current scheduling problem, the tabular method will suffer from the curse of dimensionality because of too many time steps and grids. In addition, the continuous variable E_i also makes the tabular method difficult to handle. To this end, we use neural networks to approximate the value function. Another problem is that the excessive number of EVs makes it difficult for us to maintain a value function for each EV. In this article, we assume that all the EVs are homogeneous, which is a quite common case for shared on-demand EV fleets [8], [11]. In this way, all those EVs have the same battery capacity, power consumption, and travel speed, so that they can share the same value function $V(s;\theta)$ approximated by the neural network, where θ are the parameters of the neural network. We train the network $V(s;\theta)$ to make it more like the true value function of a given policy.

Algorithm 1 shows the training process of neural network $V(s;\theta)$. To improve the stability of training, one technique is to use a target network to calculate the TD target value [5]. During initialization, we create a copy of the neural network $V(s;\theta)$, denoted as $\hat{V}(s;\hat{\theta})$. Network $\hat{V}(s;\hat{\theta})$ is called the target network, and original network $V(s;\theta)$ is called the online network. During the training process, the parameters θ of online network are updated every step, but the parameters $\hat{\theta}$ of target network are updated every C steps (making it equal to the current θ). With the target network, TD target value can be calculated as:

$$\hat{y}_{ij} = \begin{cases} r_{ij}, & s_{ij} \text{ is terminal} \\ r_{ij} + \gamma^{\Delta t_{ij}}\hat{V}\left(s_{ij};\hat{\theta}\right), & s_{ij} \text{ is non-terminal} \end{cases} \tag{15}$$

Another technique used to stabilize training and also improve data efficiency is the experience replay mechanism [29]. Specifically, we maintain a replay buffer $\mathcal{D}$ of capacity D for the entire EV fleet. After getting a dispatch assignment, each EV i will store transition (s_i, r_{ij}, s_{ij}) in the replay buffer $\mathcal{D}$. At each step of training, we randomly choose M transitions from $\mathcal{D}$ to form a mini-batch $\mathcal{M}$. With the mini-batch, we can calculate the loss function $L(\theta)$, which is defined as the mean square error (MSE) between the TD target value and the estimated state value $V(s_i;\theta)$:

$$L(\theta) = \frac{1}{M}\sum_{(s_i, r_{ij}, s_{ij})\in\mathcal{M}}\left[\hat{y}_{ij} - V(s_i;\theta)\right]^2 \tag{16}$$

The parameters θ are updated by gradient descent to minimize the loss function:

$$\theta \leftarrow \theta - lr\nabla_\theta L(\theta) \tag{17}$$

where lr is the learning rate. Since the parameters θ are continuously updated, the policy generated by the state value $V(s;\theta)$ is also continuously updated, which means that the transitions stored in the replay buffer are not generated by the same policy. This way in which the evaluated policy differs from the policy used to generate data is called *off-policy* learning.

Since continuous exploration is the key to making policy evaluation work [28], we apply ε-greedy policy to generate scheduling actions during the training process: Each grid g performs random action (randomly generate action $\boldsymbol{a}_g$ under constraints (11a)–(11f)) with probability ε, and greedy action (get action $\boldsymbol{a}_g$ by solving (11)) with probability $1-\varepsilon$. The probability ε decays from 1 to 0.1 in steps of $\Delta\varepsilon$ and then remains unchanged at 0.1. Note that when testing a trained neural network, we do not add any exploration but apply a completely greedy policy, i.e., $\varepsilon = 0$.

Similar to most popular DRL algorithms (e.g., DQN, DDPG, SAC), the proposed method applies three elements to improve performance: function approximation (i.e., neural networks) for scalability and generalization, bootstrapping for computational and data efficiency [24], and off-policy learning for experience replay. However, combining these three elements may lead to instability and divergence [30]. For this reason, we present a tabular method in the Appendix as a fallback option. The tabular method eschews function approximation and off-policy, so it is susceptible to the curse of

Algorithm 1: Policy Evaluation With Neural Networks

Initialize $\varepsilon = 1$;
Initialize replay buffer $\mathcal{D}$ with a capacity of D;
Initialize network $V(s;\theta)$ with random parameters θ;
Initialize target network $\hat{V}(s;\hat{\theta})$ with parameters $\hat{\theta} \leftarrow \theta$;
for *episode* $= 1$ **to** *number of episodes* **do**
  Receive initial state s_i for each EV i;
  for *time step* $t = 0$ **to** $T-1$ **do**
    foreach *grid* $g \in \mathcal{G}$ **do**
      foreach *EV* $i \in \mathcal{I}_g$ **do**
        foreach *dispatch* $j \in \mathcal{J}_g$ **do**
          Calculate $\Delta E_{ij}, r_{ij}, s_{ij}$;
          Calculate y_{ij} according to (10);
      Draw a sample u from uniform distribution $U(0,1)$;
      if $u < \varepsilon$ **then**
        Randomly generate $\boldsymbol{a}_g = [b_{ij}]_{i\in\mathcal{I}_g, j\in\mathcal{J}_g}$ under constraints (11a)–(11f);
      else
        Get $\boldsymbol{a}_g = [b_{ij}]_{i\in\mathcal{I}_g, j\in\mathcal{J}_g}$ by solving (11);
      foreach $b_{ij} \in \boldsymbol{a}_g$ **do**
        if $b_{ij} = 1$ **then**
          Assign dispatch j to EV i;
          Store transition (s_i, r_{ij}, s_{ij}) in $\mathcal{D}$;
    if $\varepsilon > 0.1$ **then** $\varepsilon \leftarrow \varepsilon - \Delta\varepsilon$;
    Randomly choose M transitions from $\mathcal{D}$ to form a mini-batch $\mathcal{M}$;
    Calculate target value $\hat{y}_{ij}$ according to (15);
    Calculate loss function $L(\theta)$ according to (16);
    Update parameters $\theta \leftarrow \theta - lr\nabla_\theta L(\theta)$;
    Every C steps reset $\hat{\theta} \leftarrow \theta$;

dimensionality and data inefficiency, but it can theoretically converge to a near-optimal policy. We give the specific steps and convergence proofs of the tabular method in the Appendix. Fortunately, many experiments have shown that DRL algorithms like DQN and DDPG, which have no theoretical convergence guarantees, can learn near-optimal policies [31], [32]. We will show that the proposed method also achieves near-optimal performance in an experimental comparison with the convergent tabular method (Section IV-C).

IV. Case Study

A. Experimental Settings

The study area is the center of Haikou City as shown in Fig. 4, which is approximately divided into 133 hexagonal grids. We assume that there are 20 grids equipped with chargers, which constitute three rechargeable areas (indicated by three different colors), denoted as $\mathcal{G}_1^{\text{ch}} = \{17, 46, 48, 65, 67, 80, 102\}$, $\mathcal{G}_2^{\text{ch}} = \{10, 12, 36, 51, 53, 56, 58\}$ and $\mathcal{G}_3^{\text{ch}} = \{89, 92, 105, 109, 112, 133\}$, respectively. All grids in each charging area are set to have the same electricity price. We appropriately scale the electricity price data of three

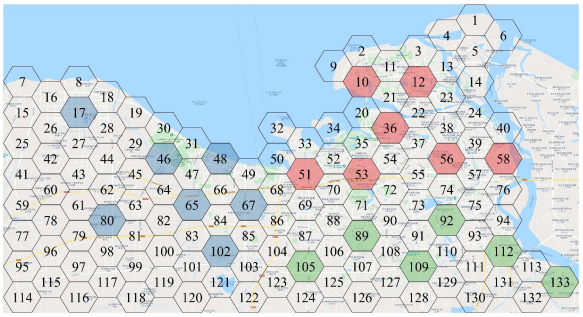


Fig. 4. The center of Haikou City divided into 133 hexagonal grids.

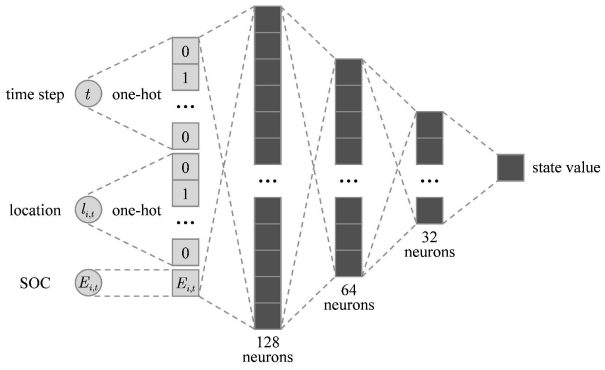


Fig. 5. Neural network architecture for estimating state value.

different buses in PJM [33] to simulate charging electricity prices. We set the number of chargers in each rechargeable grid $N_g^{\text{ch}} = 12$, and the power of each charger $P_{\text{ch}} = 30$kW. The total number of EVs is set to 800. The characteristics of EVs are modeled after Beiqi New Energy EU300 with a battery capacity of 45kWh and a range of 300km. The initial EV position is randomly placed, and the initial SOC bounded between 0.2 and 0.8 is sampled from $\mathcal{N}(0.5, 0.1^2)$ [19]. Each time step is set to 15 minutes, but higher accuracy can be used in actual scheduling. Considering that a charging behavior may affect the rewards for more than one day, we set each episode to one week, i.e., $T = 672$.

The data provided by Didi Chuxing [9] includes on-demand orders in the center area of Haikou City for 26 consecutive weeks (2017/05/01–2017/10/29). The order information includes order generation time, price, origin, destination and duration. We randomly select about 1.6×10^5 orders at a time from the dataset and compose an episode (a week) based on the order generation time (day of week, time of day). Notice that the random seeds used in the training phase and the testing phase are different. We set that each order will be cancelled if it is not serviced within half an hour after it is generated. Fig. 6 shows the cumulative order requests for the entire study area within a week. It can be seen that there is more demand for orders on the right side of the area. Combined with the distribution of chargeable areas, we can find that $\mathcal{G}_2^{\text{ch}} > \mathcal{G}_3^{\text{ch}} > \mathcal{G}_1^{\text{ch}}$ in terms of order demand. Fig. 7 shows the change in order demand over time in the entire study area. It can be seen that 14:00–19:00 is the peak period of order demand every day.

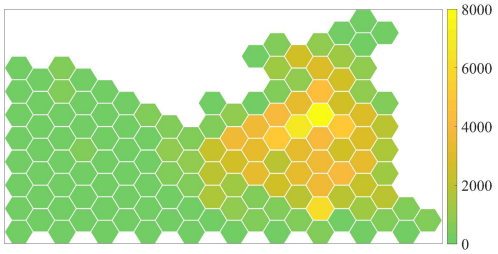


Fig. 6. Number of order requests for different locations during the week.

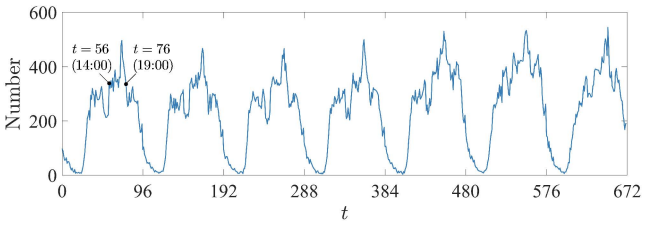


Fig. 7. Number of order requests per time step in the study area.

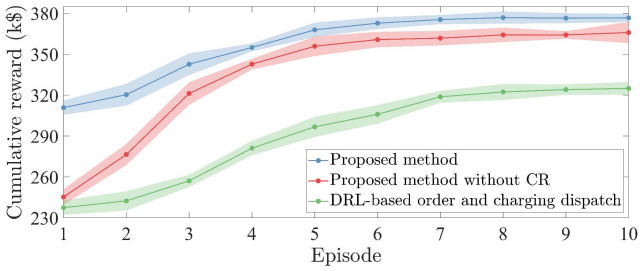


Fig. 8. Episodic average cumulative reward of EV fleet over 6 different random seeds during training. Error bars are the 95% confidence intervals.

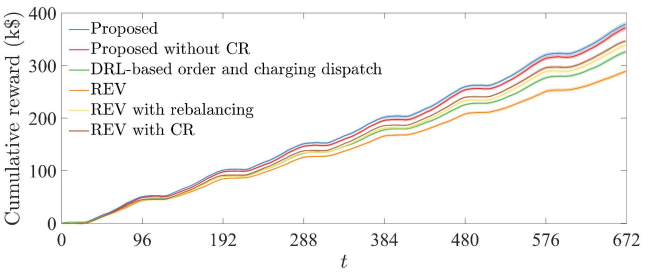


Fig. 9. Average cumulative reward of EV fleet over 6 different random seeds during testing.

The neural network used to approximate the state value function is shown in Fig. 5, which has 3 hidden layers with 128, 64, and 32 neurons respectively. We apply rectified linear unit (ReLU) [34] to each layer. At the input layer, the discrete state variables (i.e., time step t and location $l_{i,t}$) are represented with one-hot encoding. The Adam optimizer [35] is used for learning the neural network parameters with a learning rate $lr = 0.001$. The other hyperparameters during training are as follows: the number of episodes is 10, the capacity of replay buffer $D = 5 \times 10^5$, the discount factor $\gamma = 0.9$, and the decay step size of the exploration probability $\Delta\varepsilon = 1.5 \times 10^{-4}$.

The BLP problem is written and solved in Python with Gurobi [36], and the neural networks are trained in Python with Pytorch [37], an open source deep learning platform. All experiments are carried out on a computer with a 10-core 3.70 GHz Intel Core i9-10900X processor and 32 GB of RAM.

B. Performance Comparison

We verified the performance of proposed method by comparing with several benchmark algorithms:

- **Revenue-based method (REV):** This method is often used as a benchmark for order dispatching algorithms, meaning that orders with higher prices will be given priority to get dispatched first [10], [13]. In order to apply to the EV fleet in this article, we set that each EV will be charged when there is no order in the current grid and there is a charger available, otherwise it will stay in the current grid.
- **REV with rebalancing:** Based on REV, we set that when the current grid has no orders and no charger is available, EV will move to the neighboring grid with a larger demand-supply gap.
- **REV with CR:** Based on REV with rebalancing, we constrain the number of EVs rebalanced to other grids as in (11f).
- **DRL-based order and charging dispatch:** This method is a state-of-the-art DRL-based EV fleet scheduling algorithm [11]. The method was proposed without taking into account the fact that the number of chargers is limited, so in this article there will be a queue of EVs waiting for charging.
- **Proposed method without CR:** The only difference from the proposed method is that we do not constrain the number of EVs rebalanced to other grids.
- **Proposed method:** Our proposed model as detailed in Section III.

The training results of the three DRL-based methods are shown in Fig. 8. One can seen that the use of CR significantly improves the efficiency of exploration (the performance of the initial episodes are improved) and increases the cumulative reward of final convergence. In addition, the method without rebalancing (i.e., DRL-based order and charging dispatch) performed worse than the other two methods.

The test results of all methods are shown in Fig. 9. One can seen that the cumulative reward of proposed method is higher than other methods. In addition, the methods without rebalancing (REV, DRL-based order and charging dispatch) are relatively low, while the methods with CR (REV with CR, proposed method) are higher than the methods with conventional rebalancing (REV with rebalancing, proposed method without CR).

The one-day trajectory of an EV during the testing phase of different methods is shown in Fig. 10. The starting location for this EV is the lower left corner of the map where order demand is low. Under the proposed method (Fig. 10(a)), the EV is gradually shifted from the lower left corner to a higher demand area for more order revenue by performing a series of rebalancing dispatches. For the proposed method without CR (Fig. 10(b)), the EV can be rebalanced to areas

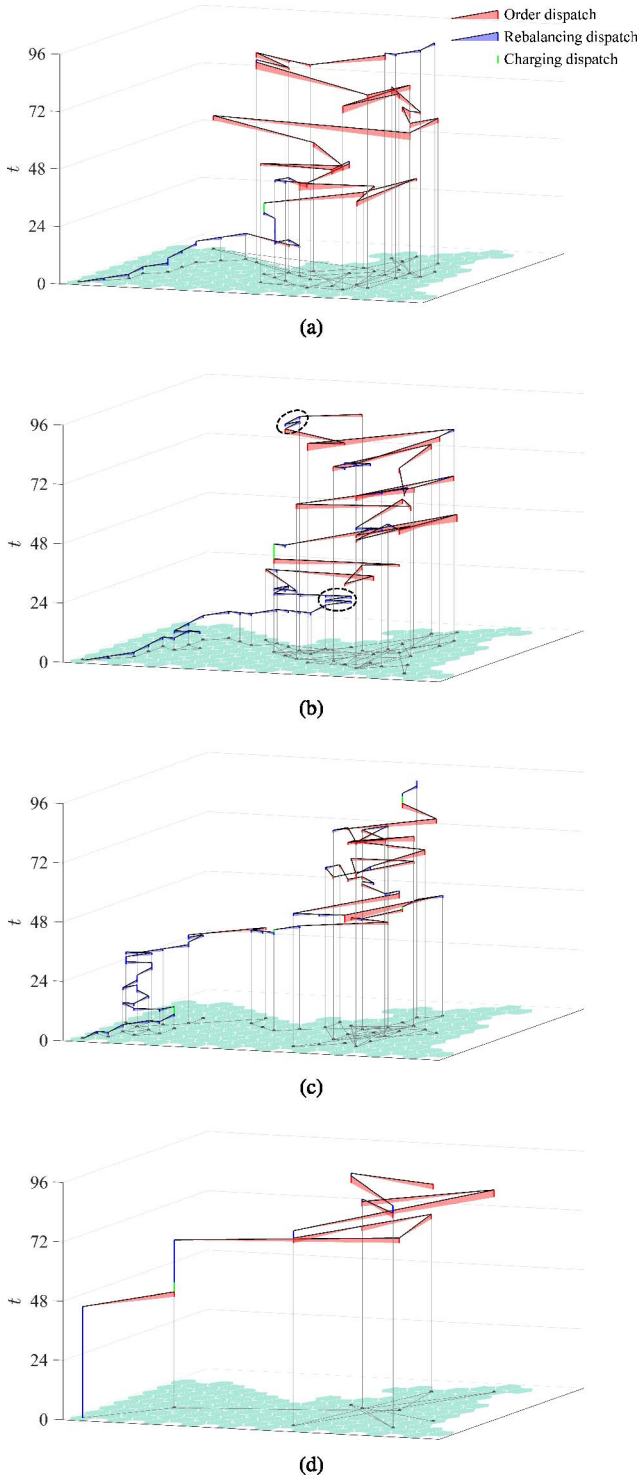


Fig. 10. The one-day trajectory of an EV during the testing phase of different methods. (a) Proposed method; (b) Proposed method without CR; (c) REV with CR; (d) DRL-based order and charging dispatch.

of high demand but sometimes the rebalancing is repeated between adjacent grids (black dashed circles), resulting in a waste of energy. For REV with CR (Fig. 10(c)), the EV can also be rebalanced to areas of high demand, but this process is circuitous due to the lack of guidance from the state value function (see Fig. 15 for state values at different locations). As for DRL-based order and charge dispatch (Fig. 10(d)), the

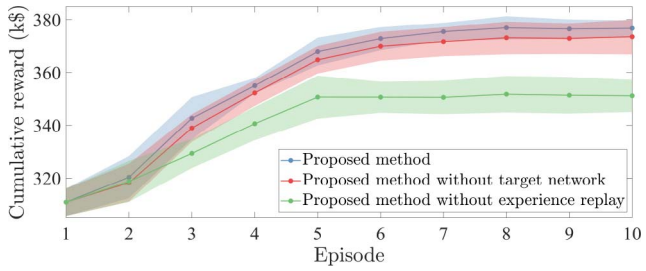


Fig. 11. Impact of the target network and experience replay mechanism on the proposed method. Error bars are the 95% confidence intervals across 6 random seeds.

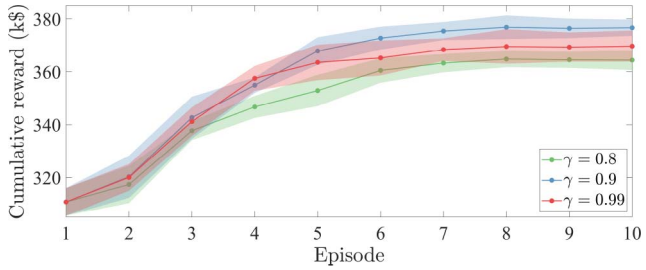


Fig. 12. Impact of the discount factor on the proposed method. Error bars are the 95% confidence intervals across 6 random seeds.

lack of vehicle rebalancing prevents the EV from proactively shifting to high-demand areas for additional revenue.

C. Stability of Training

This subsection analyzes the factors that affect the training stability of the proposed method, including the target network, experience replay mechanism, and discount factor. In addition, comparison with a convergent tabular method is used to reflect the near-optimal performance achieved by the proposed method.

Fig. 11 shows the impact of the target network and experience replay mechanism on the proposed method. The absence of experience replay significantly reduces the average cumulative reward (by 7.05%), while absence of target network reduces it not so much (by 0.86%). The absence of any of these increases the confidence interval, i.e., makes the training unstable.

Fig. 12 illustrates the impact of the discount factor γ on the performance of the proposed method. A low discount factor can cause agent to prioritize excessively immediate rewards and become myopic to future rewards [24]; however, targeting a high discount factor may lead to instability or divergence in the state value function estimates, yielding a poor quality policy [38]. Fig. 12 shows that when $\gamma = 0.9$, the proposed method achieves an effective tradeoff between the above two effects, converging to a more profitable policy.

We compare the proposed method with the tabular method provided in the Appendix. In the tabular method setting, we discretize the SOC into 11 levels, i.e., $E_{i,t} \in \{0, 0.1, 0.2, \ldots, 1\}$, and set the thresholds $\delta = 2.0$ and $\delta = 1.5$, respectively. The threshold δ determines the accuracy of the policy evaluation. Fig. 13 shows the experimental results. It can be seen that the smaller the threshold δ, the

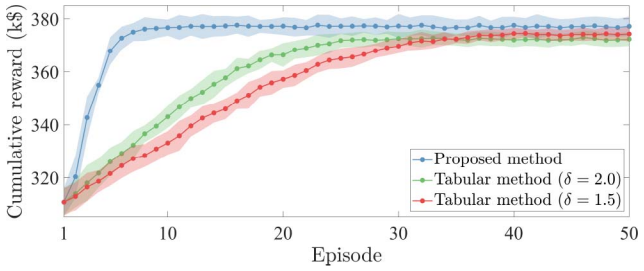


Fig. 13. Episodic average cumulative reward of EV fleet for the proposed method and tabular method. Error bars are the 95% confidence intervals across 6 random seeds.

TABLE I
COMPUTATIONAL TIME TO REACH CONVERGENCE OF THE PROPOSED METHOD AND THE TABULAR METHOD

Method	Total time (min)	Number of episodes	Average time per episode (min)
Proposed method	177.38	10	17.74
Tabular method ($\delta = 2.0$)	321.29	30	10.71
Tabular method ($\delta = 1.5$)	575.82	50	11.52

slower the convergence of the tabular method, but the higher the cumulative reward after convergence. In addition, the tabular method has a smaller confidence interval, i.e., higher stability, but converges significantly slower than the proposed method. Table I also shows that the total computational time required for the tabular method to reach convergence exceeds that of the proposed method. The slow convergence is due to the fact that the tabular method only updates one state value estimate per update, so EVs need to visit each state and try each action multiple times to get a good estimate. Therefore, the tabular method, although convergent, is computationally and data inefficient and can be used as a fallback option in case the proposed method is not convergent. Fortunately, Fig. 13 shows that the proposed method converges well and outperforms the tabular method in terms of average cumulative rewards over 50 episodes. The tabular method has been proven to converge to a near-optimal policy (Appendix), which means that the proposed method also achieves a near-optimal performance.

D. State Value Analysis

We analyze the changes in state value over time, SOC, and location based on trained neural network. Fig. 14 shows the state value in grid 55 as a function of time and SOC. Fig. 14(a) reflects that the change in state value is similar every day: it is higher before the peak demand period and lower before the peak charge period (at about $t = 96$, i.e., 24:00). Figs. 14(a) and (b) both reflect a positive correlation between state value and SOC. In addition, comparing curve $t = 60$ (15:00) and curve $t = 66$ (17:30) in Fig. 14(b), it can be found that when SOC > 0.2, the state value is similar, but as SOC approaches 0, the downward trend of curve $t = 66$ (17:30) is more obvious. This is because it is only 1.5 hours from 17:30 to the end of the peak demand (as shown in Fig. 7), so the EV with small SOC will lose this important order revenue because it needs to be charged first.

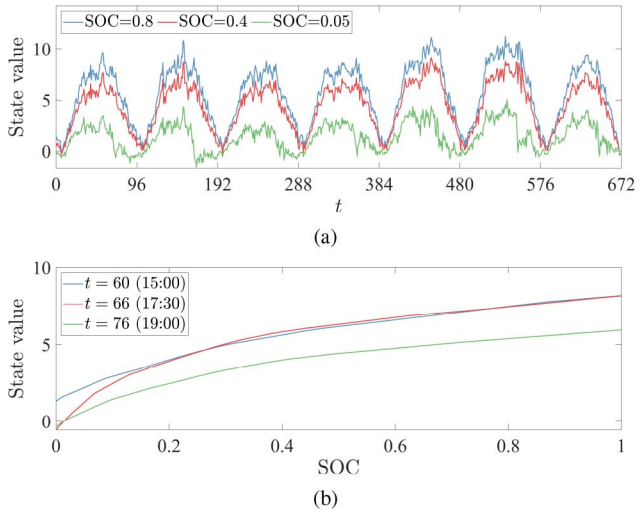


Fig. 14. Change in state value of grid 55 over time and SOC. (a) Time; (b) SOC.

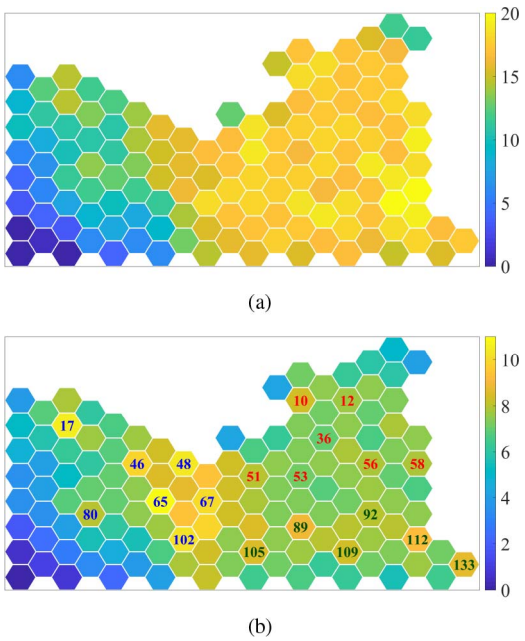


Fig. 15. State value at different locations. (a) SOC = 0.8; (b) SOC = 0.05.

We next analyze the state value in different locations. Fig. 15 shows the state value of two different SOCs at 08:00 on Thursday. It can be seen that when SOC = 0.8, the state value near the high-demand area is higher, but when SOC = 0.05, the rechargeable grids, especially the grids with lower electricity prices (see Fig. 16(a) for charging prices), have higher state value.

E. Impact of Different Electricity Pricing Mechanisms

We analyze the impact of four different electricity pricing mechanisms on charging behavior:

- Price varies both with time and location. This is the default setting. As described in Section IV-A, the charging price changes every time step and varies from one rechargeable area to another.

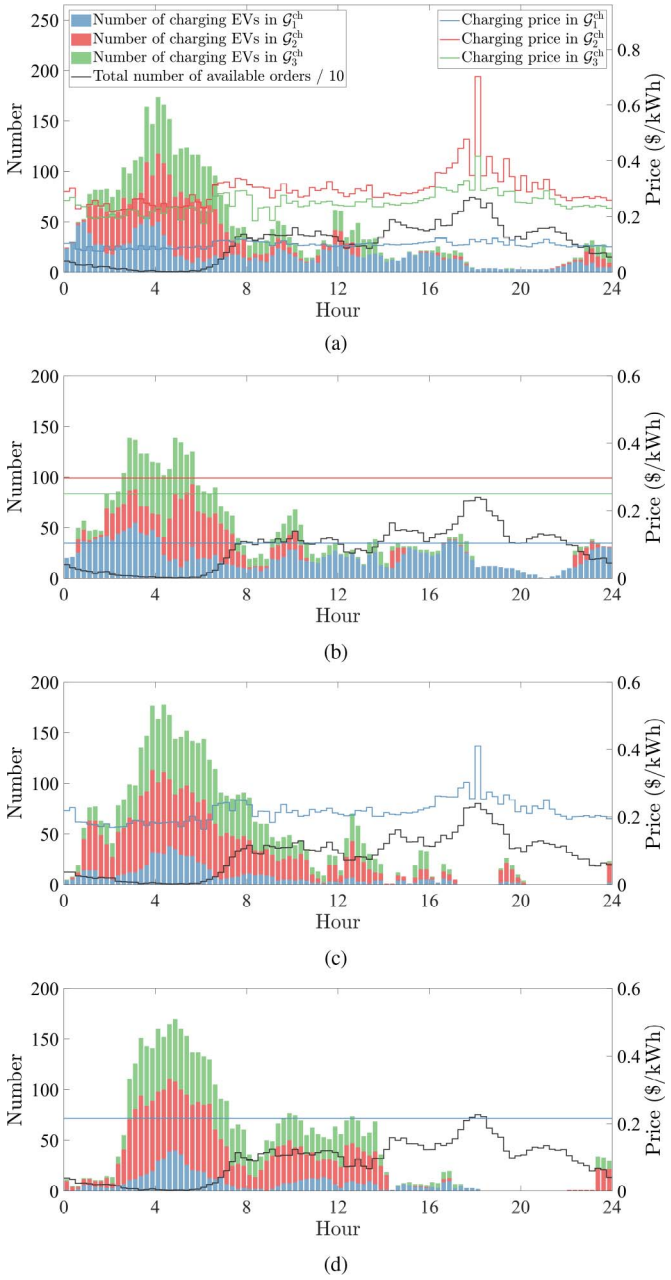


Fig. 16. The variation in the number of charging EVs during a typical day for different pricing mechanisms. (a) Price varies both with time and location; (b) Price does not vary with time but with location; (c) Price varies with time but not with location; (d) Price varies neither with time nor with location.

- Price does not vary with time but with location. The new charging price is the average price of all time steps in one day.
- Price varies with time but not with location. The new charging price is the average price for the entire study area.
- Price varies neither with time nor with location. The new charging price is the average price of all time steps throughout the study area in one day.

The variation in the number of charging EVs during a typical day for different pricing mechanisms is shown in Fig. 16. In order to analyze the distribution of charging behavior, we also plotted the total number of available orders and the electricity price of each rechargeable area during this period. For prices that vary with time and location, Fig. 16(a) shows that in the early morning hours when electricity prices and demand are lower, there are more EVs charged, while in the afternoon and evening hours when electricity prices and demand are higher, fewer EVs are charged. In addition, it can be seen that the number of charging EVs in area $\mathcal{G}_1^{ch}$ is generally the largest. This shows that although area $\mathcal{G}_1^{ch}$ is far from the high-demand area, its lower electricity price still attracts a large number of EVs by choosing a series of movable dispatches (order or rebalancing dispatch) to be recharged here. Comparing Figs. 16(a) and (b), it can be seen that the charging behavior does not change much after the price change cycle becomes one day. This is because the impact of price and demand on charging behavior is similar. For example, the order demand and electricity price every afternoon are high, and any of these factors will hinder EV charging. Figs. 16(c) and (d) reflect that when the price of the entire area is the same, the number of EVs selected to charge in area $\mathcal{G}_1^{ch}$ is greatly reduced, which is because area $\mathcal{G}_1^{ch}$ is far from the high-demand area. In addition, Fig. 16(d) reflects that the charging distribution is no longer affected by price fluctuations and the randomness is reduced.

V. Conclusion

In this article, we model the joint optimization framework combining charging scheduling, order dispatching and vehicle rebalancing for large-scale shared on-demand EV fleet operator as a POMDP, and develop a near-optimal solution method based on DRL and BLP. A simulation experiment based on city-scale real-world data from Haikou City shows that the overall profit can be significantly increased by coordinating those scheduling decisions and dynamically responding to order demand and charging prices, such as rebalancing EVs to high demand areas (or low electricity price areas) to increase order revenue (or reduce charging costs). In addition, the proposed constrained rebalancing method significantly improves the exploration efficiency of training and can improve the performance of conventional policies. Moreover, we provide a tabular method with proved convergence as a fallback option to demonstrate the near-optimal characteristics of the proposed approach.

Future work can be model the scheduling process in more detail: For order dispatching, we will consider that an EV can simultaneously carry multiple passengers with different destinations; for charging scheduling, we will consider that EVs can be discharged to the grid for participating power market. In addition, we will consider benefits to participants other than EV fleet operator, such as passenger waiting costs, to maximize social welfare.

Appendix
Convergent Tabular Method

We provide a tabular method for the EV fleet scheduling problem as shown in Algorithm 2. Due to the use of tables, we have to discretize the continuous state variable SOC, which

Algorithm 2: Tabular Method for EV Fleet Scheduling

For all $s \in \mathcal{S}$, $V(s) \leftarrow 0$, $V'(s) \leftarrow \infty$;

Policy Evaluation:
For all $s \in \mathcal{S}$, $N(s) \leftarrow 0$;
while $\|V' - V\|_\infty > \delta$ **do**
  $V' \leftarrow V$;
  for *time step* $t = 0$ **to** $T-1$ **do**
    foreach $a_i \in \pi(\boldsymbol{s}_g)$ **do**
      Take action a_i, observe r_{ij} and s_{ij};
      $N(s_i) \leftarrow N(s_i) + 1$;
      $V(s_i) \leftarrow V(s_i) + \frac{1}{N(s_i)}[r_{ij} + \gamma^{\Delta t_{ij}} V(s_{ij}) - V(s_i)]$;

Policy Improvement:
policy-stable $\leftarrow$ *true*;
foreach $\boldsymbol{s}_g \in \mathcal{S}_g$ **do**
  old-action $\leftarrow \pi(\boldsymbol{s}_g)$;
  $\pi(\boldsymbol{s}_g) \leftarrow \arg\max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} [r_{ij} + \gamma^{\Delta t_{ij}} V(s_{ij})]$;
  if *old-action* $\neq \pi(\boldsymbol{s}_g)$ **then** *policy-stable* $\leftarrow$ *false*;
if *policy-stable* **then** stop and return V and π **else** go to 2;

is often used in previous methods [1], [20]. In this way, the scheduling problem satisfies the definition of a *finite* MDP: the sets of states, actions, and rewards all have a finite number of elements. In addition, the tabular setting no longer relies on the experience replay mechanism, which is used to stabilize the training of neural networks but requires off-policy learning. Based on these, we will prove the convergence of the tabular method.

In order to facilitate the proof, we first describe the joint scheduling problem from the perspective of the entire fleet: At each time step t the EV fleet receives a global state $\mathbf{s}_t \in \mathcal{S}_\mathcal{G}$, takes an action $\mathbf{a}_t = [a_i]_{i \in \bigcup_g \mathcal{I}_g^+}$ and receives a scalar reward $\mathbf{r}_t = \sum_g \sum_{i \in \mathcal{I}_g^+} r_{i,t}$, where $\mathcal{I}_g^+$ denotes the set of all EVs (including available and unavailable EVs) in grid g.

We start from introducing the assumptions:

Assumption 1: Each state is visited infinitely often.

Assumption 2: The state value function of the entire EV fleet $\mathbf{V}_\pi(\mathbf{s})$ equals the summation of the individual EVs' state value functions, i.e., $\mathbf{V}_\pi(\mathbf{s}) = \sum_g \sum_{i \in \mathcal{I}_g^+} V_\pi(s_i)$.

Assumption 3: All EVs are homogeneous, i.e., they have same the battery capacity, power consumption, and travel speed, so that they can share the same state value function $V(s)$.

Note that Assumptions 2 and 3 have been used to address the curse of dimensionality of dealing with the entire fleet's state domain [8], [11], [13]. We have also used these two assumptions in our proposed DRL-based method: Assumption 2 for formulating a decentralized scheduling policy (Section II-E) and Assumption 3 for sharing the neural network (Section III-B).

Our proof is also built upon the two lemmas as follows:

Lemma 1: The random process $\{\Theta_n\}$ defined in $\mathbb{R}$ as

$$\Theta_{n+1}(x) = (1 - \alpha_n(x))\Theta_n(x) + \alpha_n(x)F_n(x)$$

converges to zero with probability 1 (w.p.1) under the following assumptions: leftmargin=*

1) The state space is finite;
2) $0 \le \alpha_n(x) \le 1$, $\sum_n \alpha_n(x) = \infty$, and $\sum_n \alpha_n^2(x) < \infty$;
3) $\|\mathbb{E}[F_n(x)|\mathcal{F}_n]\|_W \le \gamma \|\Theta_n\|_W + c_n$, where $\gamma \in [0, 1)$ and c_n converges to zero w.p.1;
4) $\mathbf{var}[F_n(x)|\mathcal{F}_n] \le K(1 + \|\Theta_n\|_W^2)$ with constant $K > 0$.

Here $\mathcal{F}_n$ denotes the filtration of an increasing sequence of σ-fields including the history of processes; $\alpha_n, \Theta_n, F_n \in \mathcal{F}_n$ and $\|\cdot\|_W$ is a weighted maximum norm [39].

Proof: See [40, Th. 1] and [41, Corollary 5] for detailed derivation. ■

Lemma 2: Let π and π' be any pair of deterministic policies such that, for all $\mathbf{s} \in \mathcal{S}_\mathcal{G}$,

$$\pi'(\mathbf{s}) = \arg\max_{\mathbf{a}} \mathbb{E}\big[\mathbf{r}_t + \gamma \mathbf{V}_\pi(\mathbf{s}_{t+1})|\mathbf{s}, \mathbf{a}\big]$$

Then $\mathbf{V}_{\pi'} \ge \mathbf{V}_\pi$. Moreover if π is not optimal, strict inequality holds in this inequality for at least one state.

Proof: See [24]. ■

For the policy evaluation in Algorithm 2, we define the n-th updated value of $\frac{1}{N(s)}$ to be $\alpha_n(s)$, then $\alpha_n(s) = \frac{1}{n}$. We next prove that the policy evaluation is convergent:

Theorem 1: For any finite MDP with a given policy π and the current estimated value function $V(s)$, the policy evaluation algorithm given by

$$V_{n+1}(s_i) = V_n(s_i) + \alpha_n(s_i)\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n\big(s_{ij}\big) - V_n(s_i)\big] \quad (18)$$

converges to the true value function $V_\pi(s)$ w.p.1 provided $\alpha_n(s) = \frac{1}{n}$ and $\gamma \in [0, 1)$.

Proof: We start by rewriting (18) as

$$V_{n+1}(s_i) = (1 - \alpha_n(s_i))V_n(s_i) + \alpha_n(s_i)\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n\big(s_{ij}\big)\big]$$

To apply Lemma 1 we subtract $V_\pi(s)$ from both sides of the above equation. If we write $\Theta_n(s) = V_n(s) - V_\pi(s)$ and $F_n(s) = r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij}) - V_\pi(s)$, we have $\Theta_{n+1}(s) = (1 - \alpha_n(s))\Theta_n(s) + \alpha_n(s)F_n(s)$.

The harmonic series is a divergent infinite series, i.e., $\sum_{n=1}^{\infty} \frac{1}{n} = \infty$. Notice that since $\alpha_n(s) = \frac{1}{n} \le 1$, $\sum_n \alpha_n(s) = \infty$ requires that all states be visited infinitely often. $\sum_n \alpha_n^2(s) = \sum_n \frac{1}{n^2}$ is convergent with an exact sum of $\frac{\pi}{6}$ [42].

The *error reduction property* of multi-step returns [24] guarantees that for any $\Delta t_{ij} \ge 1$,

$$\begin{aligned}
&\max_s |\mathbb{E}_\pi[F_n(s_i)|s_i = s]| \\
&\quad = \max_s \big|\mathbb{E}_\pi\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n\big(s_{ij}\big)|s_i = s\big] - V_\pi(s)\big| \\
&\quad \le \gamma^{\Delta t_{ij}} \max_s |V_n(s) - V_\pi(s)|
\end{aligned}$$

which means the multi-step look-ahead reward gives a smaller worst-case error in the estimation of the true value function. It is now immediate from the above formula that

$$\begin{aligned}
\|\mathbb{E}[F_n(s)|\mathcal{F}_n]\|_\infty &\le \gamma^{\Delta t_{ij}} \|V_n(s) - V_\pi(s)\|_\infty \\
&\le \gamma \|V_n(s) - V_\pi(s)\|_\infty = \gamma \|\Theta_n\|_\infty
\end{aligned}$$

Finally,

$$\mathbf{var}[F_n(s)|\mathcal{F}_n] = \mathbb{E}\Big[(F_n(s) - \mathbb{E}[F_n(s)])^2|\mathcal{F}_n\Big]$$

$$
\begin{aligned}
&= \mathbb{E}\Big[\big(r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij}) - V_\pi(s) \\
&\qquad - \mathbb{E}\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij})\big] + V_\pi(s)\big)^2 | \mathcal{F}_n\Big] \\
&= \mathbb{E}\Big[\big(r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij}) - \mathbb{E}\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij})\big]\big)^2 | \mathcal{F}_n\Big] \\
&= \mathbf{var}\big[r_{ij} + \gamma^{\Delta t_{ij}} V_n(s_{ij}) | \mathcal{F}_n\big]
\end{aligned}
$$

which, due to the fact that r_{ij} is bounded, clearly verifies

$$\mathbf{var}[F_n(s)|\mathcal{F}_n] \le K\Big(1 + \|\Theta_n\|_W^2\Big)$$

for some constant K.

Then, by Lemma 1, Θ_n converges to zero w.p.1, i.e., $V_n(s)$ converges to $V_\pi(s)$ w.p.1. ∎

We finally prove the convergence of Algorithm 2:

Theorem 2: Algorithm 2 converges to an optimal policy under Assumptions 1, 2, and 3.

Proof: The goal of the joint optimization problem should be finding the optimal policy to maximize the expected total return of EV fleet, i.e., state value $\mathbf{V}_\pi(\mathbf{s})$. According to Lemma 2, maximizing $\mathbb{E}[\mathbf{r}_t + \gamma \mathbf{V}_\pi(\mathbf{s}_{t+1})|\mathbf{s}, \mathbf{a}]$ can achieve non-decreasing $\mathbf{V}_\pi(\mathbf{s})$, and we decompose $\max_{\mathbf{a}} \mathbb{E}[\mathbf{r}_t + \gamma \mathbf{V}_\pi(\mathbf{s}_{t+1})|\mathbf{s}, \mathbf{a}]$ as follows:

$$
\begin{aligned}
&\max_{\mathbf{a}} \mathbb{E}\big[\mathbf{r}_t + \gamma \mathbf{V}_\pi(\mathbf{s}_{t+1})|\mathbf{s}, \mathbf{a}\big] && \text{(19a)} \\
&= \max_{\mathbf{a}} \sum_g \sum_{i \in \mathcal{I}_g^+} \mathbb{E}\big[r_{i,t} + \gamma V_\pi\big(s_{i,t+1}\big)|s_i, a_i\big] && \text{(19b)} \\
&= \sum_g \max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g^+} \mathbb{E}\big[r_{i,t} + \gamma V_\pi\big(s_{i,t+1}\big)|s_i, a_i\big] && \text{(19c)} \\
&= \sum_g \max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} \mathbb{E}\big[r_{i,t} + \gamma V_\pi\big(s_{i,t+1}\big)|s_i, a_i\big] \\
&\quad + \sum_g \sum_{i \in \mathcal{I}_g^-} \big[r_{i,t} + \gamma V_\pi\big(s_{i,t+1}\big)\big] && \text{(19d)}
\end{aligned}
$$

where $i \in \mathcal{I}_g^-$ denotes the set of unavailable EVs (in service or charging) in grid g. Based on Assumptions 2 and 3, we can rewrite (19a) as (19b). Since only available action for each EV $i \in \mathcal{I}_g^-$ is to continue to execute its current dispatch, we can decompose (19c) into (19d), and only need to maximize the first item of (19d). For the same reason, deciding the action of EV $i \in \mathcal{I}_g$ at the next time step is equivalent to deciding which dispatch the EV will perform next, i.e., $\max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} \mathbb{E}[r_{i,t} + \gamma V_\pi(s_{i,t+1})|s_i, a_i] = \max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} \mathbb{E}[r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij})|s_i, a_i]$. In the current scheduling problem, the reward r_{ij} and the new state s_{ij} obtained after taking action a_i can be determined in advance, which means that the expected value is still equal to itself, i.e., $\max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} \mathbb{E}[r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij})|s_i, a_i] = \max_{\boldsymbol{a}_g} \sum_{i \in \mathcal{I}_g} [r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij})]$.

In summary, maximizing $\sum_{i \in \mathcal{I}_g} [r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij})]$ for each grid is equivalent to maximizing $\mathbb{E}[\mathbf{r}_t + \gamma \mathbf{V}_\pi(\mathbf{s}_{t+1})|\mathbf{s}, \mathbf{a}]$. According to Lemma 2, we can conclude that for all $s \in \mathcal{S}$, $\pi'(s) = \sum_{i \in \mathcal{I}_g} [r_{ij} + \gamma^{\Delta t_{ij}} V_\pi^i(s_{ij})]$, then $\mathbf{V}_{\pi'} \ge \mathbf{V}_\pi$. Policy π' is guaranteed to be a strict improvement over π (unless π is already optimal). Because a finite MDP has only a finite number of policies [24], Algorithm 2 must converge to an optimal policy in a finite number of iterations. ∎

Noting that Assumption 1 is difficult to implement in practice, we use a small threshold δ, as shown in Algorithm 2, to determine the accuracy of policy evaluation. Therefore, Algorithm 2 converges to a near-optimal policy in practice.

ACKNOWLEDGMENT

The authors thanks for the support of data source from Didi Chuxing GAIA Initiative.

REFERENCES

[1] F. Rossi, R. Iglesias, M. Alizadeh, and M. Pavone, "On the interaction between autonomous mobility-on-demand systems and the power network: Models and coordination algorithms," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 384–397, Jan. 2019.

[2] R. Zhang, F. Rossi, and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 1382–1389.

[3] L. Duan, Y. Wei, J. Zhang, and Y. Xia, "Centralized and decentralized autonomous dispatching strategy for dynamic autonomous taxi operation in hybrid request mode," *Transp. Res. C Emerg. Technol.*, vol. 111, pp. 397–420, Jan. 2020.

[4] M. Tsao, R. Iglesias, and M. Pavone, "Stochastic model predictive control for autonomous mobility on demand," in *Proc. IEEE 21st Int. Conf. Intell. Transport. Syst. (ITSC)*, 2018, pp. 3941–3948.

[5] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[6] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[7] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[8] Z. Xu *et al.*, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2018, pp. 905–913.

[9] *Data Source: Didi Chuxing GAIA Initiative*. Accessed: Nov. 17, 2019. [Online]. Available: https://gaia.didichuxing.com

[10] J. Jin *et al.*, "CORIDE: Joint order dispatching and fleet management for multi-scale ride-hailing platforms," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1983–1992.

[11] J. Shi, Y. Gao, W. Wang, N. Yu, and P. A. Ioannou, "Operating electric vehicle fleet for ride-hailing services with reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 21, 2019, doi: 10.1109/TITS.2019.2947408.

[12] X. Tang *et al.*, "A deep value-network based approach for multi-driver order dispatching," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2019, pp. 1780–1790.

[13] M. Li *et al.*, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *Proc. World Wide Web Conf.*, 2019, pp. 983–994.

[14] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[15] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," 2018. [Online]. Available: arXiv:1802.05438.

[16] K. Lin, R. Zhao, Z. Xu, and J. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2018, pp. 1774–1783.

[17] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, Mar. 2015.

[18] T. Ding, Z. Zeng, J. Bai, B. Qin, Y. Yang, and M. Shahidehpour, "Optimal electric vehicle charging strategy with Markov decision process and reinforcement learning technique," *IEEE Trans. Ind. Appl.*, vol. 56, no. 5, pp. 5811–5823, Sep./Oct. 2020.

[19] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5246–5257, Nov. 2018.

[20] B. Turan, R. Pedarsani, and M. Alizadeh, "Dynamic pricing and management for electric autonomous mobility on demand systems using reinforcement learning," 2019. [Online]. Available: arXiv:1909.06962.

[21] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: arXiv:1509.02971.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018. [Online]. Available: arXiv:1801.01290.
[23] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Mach. Learn.*, 1994, pp. 157–163.
[24] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[25] C. P. Birch, S. P. Oom, and J. A. Beecham, "Rectangular and hexagonal grids used for observation, experiment and simulation in ecology," *Ecol. Model.*, vol. 206, nos. 3–4, pp. 347–359, 2007.
[26] J. Ke *et al.*, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4160–4173, Dec. 2018.
[27] R. S. Sutton, D. Precup, and S. Singh, "Between MDPS and semi-MDPS: A framework for temporal abstraction in reinforcement learning," *Artif. Intell.*, vol. 112, nos. 1–2, pp. 181–211, 1999.
[28] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.
[29] L.-J. Lin, "Reinforcement learning for robots using neural networks," Dept. School Comput. Sci., Carnegie–Mellon Univ., Pittsburgh, PA, USA, Rep. CMU-CS-93-103, 1993.
[30] J. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximationtechnical," Lab. Inf. Decis. Syst., Massachusetts Inst. Technol., Rep. LIDS-P-2322, 1996.
[31] Y. Liang, C. Guo, Z. Ding, and H. Hua, "Agent-based modeling in electricity market using deep deterministic policy gradient algorithm," *IEEE Trans. Power Syst.*, early access, Jun. 2, 2020, doi: 10.1109/TPWRS.2020.2999536.
[32] H. Lee, M. Girnyk, and J. Jeong, "Deep reinforcement learning approach to MIMO precoding problem: Optimality and robustness," 2020. [Online]. Available: arXiv:2006.16646.
[33] *PJM Market Data*. Accessed: Nov. 25, 2019. [Online]. Available: https://www.pjm.com/
[34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 315–323.
[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Rep. (ICLR)*, San Diego, CA, USA, May 2015, p. 6.
[36] (2019). *Gurobi*. [Online]. Available: https://www.gurobi.com/
[37] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[38] V. François-Lavet, R. Fonteneau, and D. Ernst, "How to discount deep reinforcement learning: Towards new dynamic strategies," 2015. [Online]. Available: arXiv:1512.02011.
[39] D. P. Bertsekas, "Weighted sup-norm contractions in dynamic programming: A review and some new applications," Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Rep. LIDS-P-2884, 2012.
[40] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 703–710.
[41] C. Szepesvári and M. L. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 2017–2060, 1999.
[42] T. M. Apostol, "A proof that Euler missed: Evaluating ζ (2) the easy way," *Math. Intell.*, vol. 5, no. 3, pp. 59–60, 1983.

Yanchang Liang (Student Member, IEEE) received the B.S. degree from the College of Electrical Engineering, North China Electric Power University, Baoding, China, in 2018. He is currently pursuing the M.S. degree with North China Electric Power University, Beijing, China.

His current research interests include control, optimization, reinforcement learning, with applications to power systems, and electric transportation systems.

Zhaohao Ding (Senior Member, IEEE) received the B.S. degree in electrical engineering and the B.A. degree in finance from Shandong University, Jinan, China, in 2010, and the Ph.D. degree in electrical engineering from the University of Texas at Arlington, Arlington, TX, USA, in 2015.

He is currently an Associate Professor with North China Electric Power University, Beijing, China. His research interests include power system planning and operation, power market, and electric transportation system.

Tao Ding (Senior Member, IEEE) received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 2009 and 2012, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2015. From 2013 to 2014, he was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. He is currently an Associate Professor with the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University. He has published more than 60 technical papers and authored by "Springer Theses" recognizing outstanding Ph.D. research around the world and across the physical sciences—*Power System Operation with Large Scale Stochastic Wind Power Integration*. His current research interests include electricity markets, power system economics and optimization methods, and power system planning and reliability evaluation. He received the Excellent Master and Doctoral Dissertation from Southeast University and Tsinghua University, and the Outstanding Graduate Award of Beijing City. He is an Editor of IEEE TRANSACTIONS ON POWER SYSTEMS, *IET Generation, Transmission and Distribution*, and *CSEE Journal of Power and Energy Systems*.

Wei-Jen Lee (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Arlington, Arlington, TX, USA, in 1985. In 1986, he joined the University of Texas at Arlington, where he is currently a Professor with the Department of Electrical Engineering and the Director of the Energy Systems Research Center. He has been involved in research on power flow, transient and dynamic stability, voltage stability, short circuit, relay coordination, power quality analysis, renewable energy, and deregulation for utility companies. He is a registered Professional Engineer in the State of Texas.