Incentivized Self-Rebalancing Fleet in Electric Vehicle Sharing

Yuguang Wu[a], Minmin Chen[b], Xin Wang[c*]
[a]Department of Industrial and Systems Engineering,
University of Wisconsin-Madison, Madison, WI, USA
[b]Amazon.com Services LLC, Seattle, WA, USA
[c]Department of Industrial and Systems Engineering &
Grainger Institute for Engineering,
University of Wisconsin-Madison, Madison, WI, USA

Abstract

With the rising need for efficient and flexible short-distance urban transportation, more vehicle sharing companies are offering one-way car-sharing services. Electrified vehicle sharing systems are even more effective in terms of reducing fuel consumption and carbon emission. In this article, we investigate a dynamic fleet management problem for an electric vehicle (EV) sharing system that faces time-varying random demand and electricity price. Demand is elastic in each time period, reacting to the announced price. To maximize the revenue, the EV fleet optimizes trip pricing and EV dispatching decisions dynamically. We develop a new value function approximation (VFA) with input convex neural networks (ICNNs) to generate high-quality solutions. Through a New York City case study, we compare it with standard dynamic programming methods and develop insights regarding the interaction between the EV fleet and the power grid.

Keywords: Dynamic programming; Revenue management; Vehicle sharing; Electric vehicle; Value function approximation.

1 Introduction

Emerging shared mobility services have been developed to overcome the low utilization of private vehicles. Instead of owning private cars, people pay for their rides to other drivers

*Corresponding author, email: xin.wang@wisc.edu

who share vehicles (e.g, Uber, Lyft) or rent cars from car-sharing companies (e.g., Zipcar, car2go, etc) by trips to meet their daily travel needs. Zipcar, a representative US car-sharing company, offers short-term car rental services. As of 2019, Zipcar has more than 12,000 vehicles in service distributed over 500 cities (Zipcar, 2019). Through its mobile app, members can reserve vehicles for different trip durations (from an hour to two days). Other car-sharing companies have also emerged in big cities across the globe. In addition to increasing transportation efficiency, achieving sustainable and eco-friendly transportation is another task of smart cities. Car-sharing companies incorporate hybrid and/or pure electric vehicle (EV) fleets to reduce carbon emissions.

Despite the benefits in transportation efficiency and sustainability, EV sharing is confronted with various operational difficulties. The fleet has to be dynamically rebalanced to serve the demand, which is fluctuating and unbalanced. Rebalance challenges have been seen in the context of bike-sharing, car-sharing, and other resource allocation problems. In typical fleet management cases, vehicles are rebalanced by paid staff. Demand-responsive pricing is another efficient tool besides manual rebalancing. The EV operator can set low prices (or even pay customers) for trips from oversupplied origins to undersupplied destinations, and set high prices vice versa.

For electric fleets, an additional difficulty is to maintain sufficient battery levels. Due to complicated demand scenarios, it may not be optimal to charge vehicles following myopic policies. Charging EV proactively can significantly increase the profit and service quality (He et al., 2021). The interaction with the power grid also affects the charging decision. More cost can be saved if the EV fleet can systematically plan for the possible electricity price fluctuation across the day.

In this article, we consider an EV fleet controlled by a centralized, profit-maximizing EV operator over a finite planning horizon. The EV operator dynamically decides the customer trip pricing and vehicle dispatching in each time period, facing random, time-varying electricity cost and price-elastic customer demand. We adopt a neural network

approximation method that produces good quality solutions in numerical experiments using New York City TLC data.

The contributions of this article are as follows. We present an MDP model to study the possibility of EV sharing fleets achieving self-rebalancing through origin-destination (OD) pricing. We deliver interesting managerial insights through numerical studies. For example, the demand-responsive pricing instrument produces higher operational profit, reduces manual rebalancing, and stabilizes the electricity consumption of the EV fleet. On the methodology side, we propose an input convex neural network (ICNN) method to approximately solve the EV fleet management problem. ICNN method allows both value-based and gradient-based learning. It shows high approximation accuracy in solving the EV fleet management problem of a small/medium scale.

The remainder of this article is organized as follows. In Section 2, we review the related literature. In Section 3, we present the MDP formulation of the problem. In Section 4, we provide the dispatching strategy based on value function approximation methods. Section 5 constructs a pricing strategy that is induced from the value functions. Section 6 presents the case study with insights into the EV fleet's behavior. Section 7 concludes the article.

2 Literature review

Given the need for user flexibility, two important aspects of EV operational are addressed in EV systems, the upfront infrastructure planning and vehicle relocation in operation (Hodgson, 1990; Melkote and Daskin, 2001; Yao et al., 2010). Some other work modeled both decisions jointly. Nair and Miller-Hooks (2011) generated the least-cost short-term vehicle redistribution strategy which satisfies all demand realizations with at least a certain probability. de Almeida Correia and Antunes (2012) presented a profit maximization model to depot location in one-way car-sharing systems where vehicle stock imbalance issues are addressed. Li et al. (2016) presented a Continuum Approximation (CA) model for the design of a one-way EV sharing system that determines the optimal EV sharing station

locations and the corresponding EV fleet sizes. He et al. (2017) studied the planning problem faced by the EV-sharing service providers in designing a geographical service region in which to operate the service. Chang et al. (2017) considered location design together with relocation problems for sharing a mixed fleet of cars under CO2 emission constraints.

Studies of EVs also have been focusing on the interaction of charging and operation decisions. Electric engineering models have been established to capture the pattern of the charging demand of large EV fleets (e.g., Li and Zhang (2012)). Pan et al. (2010) studied how to best site battery exchange stations in terms of how they can support both the transportation system and the power grid. Adler and Mirchandani (2014) applied a Q-learning method to optimize the online routing and battery reservations of EVs in a transportation network with battery swapping stations. Boyacı et al. (2017) solved a multi-objective mixed-integer linear programming model to optimize operational decisions for vehicle and personnel relocation in a car-sharing system. Zhang et al. (2021) studied both facility planning decisions and fleet operation decisions in EV sharing systems with vehicle-to-grid (V2G) integration via a two-stage stochastic integer program.

Models for high-level EV system planning usually oversimplifies the stochastic and dynamic nature of EV sharing operations. Another stream of literature, investigating the dynamic resource allocation problem, provide useful solution strategies in solving the dynamic EV systems. A comprehensive review of the usage of approximate dynamic programming (ADP) in resource allocation problems can be found in Powell and Topaloglu (2006). Godfrey and Powell (2002) considered a stochastic version of a dynamic resource allocation problem. To solve the intractable program, they proposed using separable piecewise linear functions to approximate the value functions in the Bellman Equation. Vehicle relocation is a special case of the generic resource allocation problem. The separable function method was then utilized in many fleet management problems Hajibabai and Ouyang (2016); Lei and Ouyang (2017). In particular, Topaloglu and Powell (2007) used the framework as a

subroutine and optimized the upper-level pricing decisions for fleet management problems. Lei et al. (2019) utilized the ADP method to solve the proposed multi-stage game-theoretic model, which addresses dynamic pricing and idling vehicle dispatching problems in ride-sharing systems. Al-Kanj et al. (2020) studied a similar dynamic planning problem of autonomous EV fleets using linear approximations and hierarchical aggregation. In this article, we extend the value function approximation method using neural networks with special architecture that guarantees input convexity. The architecture is first proposed by Amos et al. (2017) to improve neural network training and inference efficiency.

3 Model

We model the EV fleet management problem as a discrete-time Markov decision process (MDP). In each time period, the centralized EV fleet manager makes a two-stage decision, pricing and dispatching, facing environmental randomnesses, such as uncertain demand and electricity price. In this section, we first introduce the notations to describe the EV sharing network. Next, we define the random price-demand relationship for customer trips. Then, we formally present the states, decisions, and transitions in the MDP. Lastly, we formulate the profit maximization problem. The summarized definitions of the model notations can be found in the supplemental online materials §A.

EV Sharing Network and Notations

Consider a centralized one-way EV sharing system, in which an EV fleet operator makes the pricing and vehicle dispatching decisions over a discrete finite planning horizon $\mathcal{T} = \{1, \cdots, T\}$. Let $\mathcal{J}$ be the finite set of EV charging stations, where each EV can be picked up and returned. Station $j \in \mathcal{J}$ can hold up to $\bar{k}_j$ EVs. All EVs in the fleet are assumed to be identical with finite state of charge (SoC) levels $\mathcal{E} = \{0, 1, \cdots, \bar{e}\}$. Let r_{jet} be the number of EV inside station $j \in \mathcal{J}$ with SoC level $e \in \mathcal{E}$. Denote $\boldsymbol{r_t} = \{r_{jet}\}_{j\in\mathcal{J}, e\in\mathcal{E}}$ as the fleet state vector. In this article, we would like to focus on a large scale, spatially dense EV fleet. Therefore, we adopt a continuous formulation of EV flows, so r_{jet} is defined

over the real-valued interval $\left[0, \bar{k}_j\right]$.

In the base model, we assume the travel time between any pair of stations is one unit time period (Godfrey and Powell, 2002). Thus, at the beginning of each time period, all EVs are settled down in the stations. We present an extended model that incorporates multi-period travels in the supplemental online materials §E.

We adopt a general definition of "travel lanes" to characterize the possible duties for EVs in each period. A lane represents either a physical EV movement between stations, in the form of *demand-serving or rebalancing*, or a non-moving behavior, in the form of *charging* or *idling*. Each lane l is associated with four attributes, the origin station o_l, the destination station d_l, the start-end SoC difference ε_l, and the revenue/cost p_{lt} if an EV travels along the lane. We then explain the physical meaning of the attributes for the specific types of lanes.

The set of *demand-serving* lanes is denoted as $\mathcal{L}_D$. For each $l \in \mathcal{L}_D$, o_l and d_l stand for the origin-destination pair; ε_l (< 0) is the required energy to complete the corresponding EV trip; p_{lt} is the EV service price. Let $\boldsymbol{p_t^D} = \{p_{lt}\}_{l \in \mathcal{L}_D}$be the price vector for demand-serving trips in period t. Similarly, we have the set of *rebalancing* lane $\mathcal{L}_R$. The only difference is that p_{lt} (< 0) represents the cost to achieve the rebalance movement of an EV from o_l to d_l in period t. The set of *charging* lanes and *idling* lanes are denoted as $\mathcal{L}_C$ and $\mathcal{L}_I$, respectively. They have $o_l = d_l$, assuming charging only can be completed in stations. ε_l (> 0) is the amount of energy charged to an EV in one period. p_{lt} (< 0) is the cost of occupying a charging spot in station o_l in period t. Similarly, we let $\varepsilon_l = 0$ for $l \in \mathcal{L}_I$, i.e., the SoC does not change for idle EVs. p_{lt} (≤ 0) is the cost of occupying a non-charging spot in station o_l. We let $\boldsymbol{p_t^{nD}} = \{p_{lt}\}_{l \in \mathcal{L}_R \cup \mathcal{L}_C \cup \mathcal{L}_I}$ be the revenue/cost vector for the non-demand-serving lanes.

The set of all lanes are defined as $\mathcal{L} = \mathcal{L}_D \cup \mathcal{L}_R \cup \mathcal{L}_C \cup \mathcal{L}_I$. Figure 1 is an illustration of an eight-lane network regarding two stations i and j.

To summarize, in each period, an EV dispatched to lane l will 1) move from o_l to d_l,

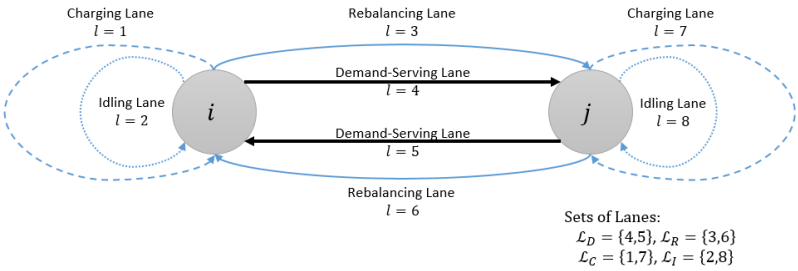


Figure 1 A sketch of the four service types

2) change its SoC from e to $e' = \min\{e + \varepsilon_l, \bar{e}\}$, and 3) contribute revenue/cost p_{lt} to the fleet. Trip pricing $\boldsymbol{p_t^D}$ is a decision controlled by the EV operator. Non-demand-serving revenue/cost $\boldsymbol{p_t^{nD}}$ are given by the environment.

Demand-Price Relation

We consider a stochastic, time-varying, and price-elastic customer demand pattern. The actual demand λ_{lt} on lane l serves as the upper bounds of the numbers of EV to be dispatched to the lane in period t. We assume EV customers have alternative transit methods so that unmet demand in each period is lost. Suppose λ_{lt} follows a linear demand-curve

$$\lambda_{lt} = (a_{lt} - b_{lt}p_{lt} + \theta_{lt})^+, \forall l \in \mathcal{L}_D \tag{1}$$

Here, $a_{lt} + \theta_{lt}$ contributes as the demand potential, $b_{lt} > 0$ is the price elasticity, and p_{lt} is the real-time EV service price. Specifically, a_{lt} and b_{lt} are information known before pricing, and θ_{lt} is a zero-mean noise known only after the price is announced.

We suppose the demand parameters a_{lt} and b_{lt} are predicted by an exogenous forecasting model. Consider a real-time demand forecasting model that passes down the parameters to the EV fleet manager at the beginning of the time period t before pricing customer trips. After the trip price p_{lt} is decided, the actual demand is observed with the noise θ_{lt}.

Exogenous Information

We define $\boldsymbol{\omega_t} = \left(\{a_{lt}, b_{lt}\}_{l\in\mathcal{L}_D}, \boldsymbol{p_t^{nD}}\right)$ as the (pre-pricing) information state. The information state contains random exogenous parameters observed at the beginning of the

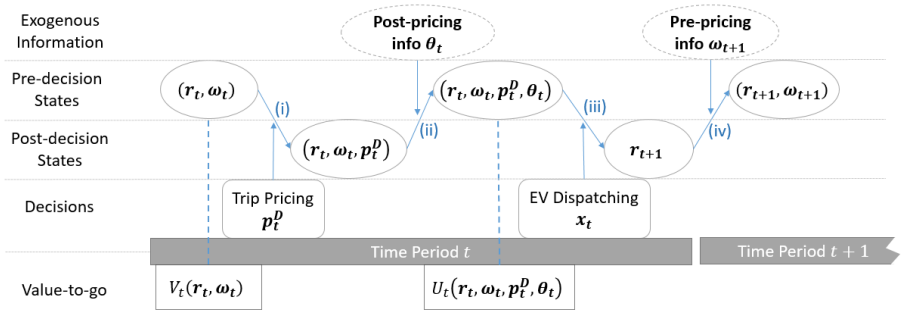


Figure 2 MDP dynamics and value-to-go functions

period, including forecasts of demand curves and random non-demand-serving costs. Note that $\boldsymbol{\omega_t}$ is defined in a generic form for notational simplicity, and not all components of $\boldsymbol{\omega_t}$ are necessarily random. The demand noise $\boldsymbol{\theta_t} = \{\theta_{lt}\}_{l \in \mathcal{L}_D}$ is referred to as the post-pricing information.

Sequence of Events and Decisions

The system alternates between pre-decision states and post-decision states illustrated by Figure 2. At the beginning of time period t, the fleet manager keeps track of the state vector $(\boldsymbol{r_t}, \boldsymbol{\omega_t})$. Then, the trip price vector $\boldsymbol{p_t^D}$ is decided and revealed to customers. Next, the post-pricing information $\boldsymbol{\theta_t}$ is observed, and the actual demand is determined by (1). Finally, each EV is dispatched to exactly one lane $l \in \mathcal{L}$ for the period. Let $x_{let} \geq 0$ be the (real-valued) number of vehicles with SoC level e dispatched to lane $l \in \mathcal{L}$, and $\boldsymbol{x_t} = \{x_{let}\}_{l \in \mathcal{L}, e \in \mathcal{E}}$ be the dispatching decision vector. All services are completed in the time period. At the end of the period, the fleet reaches a new physical state $\boldsymbol{r_{t+1}}$. Along with the $t+1$ information state $\boldsymbol{\omega_{t+1}}$, the system advances to the next time period and the decision process repeats.

In this article, we assume the probability distributions of the exogenous information are time-dependent, known and Markovian (i.e., independent of the MDP history). Practically speaking, we suppose $\boldsymbol{\omega_t}$ and $\boldsymbol{\theta_t}$ can be sampled effortlessly through offline Monte Carlo simulation. The random information could depend on the corresponding post-decision state

prior to the information revalation. That is, the distribution of $\boldsymbol{\theta_t}$ may rely on $\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}\right)$ and the distribution of $\boldsymbol{\omega_{t+1}}$ may rely on $\boldsymbol{r_{t+1}}$.

In each cycle, there are four transition steps denoted by (i) to (iv) in Figure 2. Transitions (i), (ii), and (iv) are the previous state joining by the corresponding new information (or prices). Transition (iii) captures the single-period movement of EVs, which is defined by

$$r_{j,e',t+1} = \sum_{\{(l,e)\in\mathcal{L}\times\mathcal{E}: d_l=j, \min\{\bar{e}, e+\varepsilon_l\}=e'\}} x_{let}, \forall j \in \mathcal{J}, \forall e' \in \mathcal{E}. \tag{2}$$

The dispatching decision $\boldsymbol{x_t}$ is subject to a set of constraints. Given $\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right)$, the set of possible EV dispatching decisions as well as the next fleet state are jointly constrained by the following set $\mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right)$. Note that we extensively regard $\boldsymbol{r_{t+1}}$ as a part of the dispatching decision since $\boldsymbol{r_{t+1}}$ is determined by $\boldsymbol{x_t}$ through (2).

$$\begin{aligned}
\mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) := \{(\boldsymbol{x_t}, \boldsymbol{r_{t+1}}) : & \\
\text{Constraints (2)}, & \\
\sum_{\{l\in\mathcal{L}: o_l=j\}} x_{let} = r_{jet}, \quad & \forall j \in \mathcal{J}, \forall e \in \mathcal{E}, && (3)\\
r_{je,t+1} \le \bar{k}_j, \quad & \forall j \in \mathcal{J}, && (4)\\
\sum_{e\in\mathcal{E}} x_{let} \le \left(a_{lt} + \theta_{lt} - b_{lt}p_{lt}\right)^+, \quad & \forall l \in \mathcal{L}_D, && (5)\\
x_{let} = 0, \quad & \forall l \in \mathcal{L}, \forall e + \varepsilon_l < 0, && (6)\\
x_{let} = 0, \quad & \forall l \in \mathcal{L}_C, \forall e \ge e^{\text{thr}} && (7)\\
x_{let} \ge 0, \quad & \forall l \in \mathcal{L}, \forall e \in \mathcal{E} \quad \}. && (8)
\end{aligned}$$

Constraints (2) and (3) are inbound/outbound flow conservation constraints. Constraints (4) are the station capacity constraints. Constraints (5) bound the number of demand-serving trips below the actual realized demand. (6) ensures the EVs have sufficient energy to meet the need of the corresponding trips. Constraints (7) enforce a maximum threshold SoC level e^{thr} for EVs to be charged. We assume e^{thr} is predetermined for the purpose of maintaining battery health. Finally, Constraints (8) enforce the nonnegativity.

EV Profit Maximization

The EV sharing system collects all the trip fares $\boldsymbol{p_t^D}$ and pays for costs $\boldsymbol{p_t^{nD}}$. To maximize the expected profit, the fleet manager determines the pricing rule $\boldsymbol{p_t^D} = P_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right)$ and the dispatching rule $\boldsymbol{x_t} = X_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right)$ where π denotes the overall decision policy.

$$\begin{aligned} V_1\left(\boldsymbol{r_1}\right) &= \max_\pi \mathbb{E}_{\boldsymbol{\omega},\boldsymbol{\theta}}\left[\sum_{t\in\mathcal{T}}\sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}} p_{lt}x_{let}\right] \\ \text{s.t.} \quad & \boldsymbol{p_t^D} = P_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right) \\ & \boldsymbol{x_t} = X_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) \\ & \left(\boldsymbol{x_t}, \boldsymbol{r_{t+1}}\right) \in \mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) \end{aligned} \tag{9}$$

We rewrite the EV profit maximization problem (9) as a sequence of Bellman optimality equations with the help of value-to-go functions.

$$V_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right) = \max_{\boldsymbol{p_t^D}} \mathbb{E}_{\boldsymbol{\theta_t}}\left[U_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right)\right] \tag{10}$$

$$U_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) = \max_{\boldsymbol{x_t}, \boldsymbol{r_{t+1}}} \sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}} p_{lt}x_{let} + \mathbb{E}_{\boldsymbol{\omega_{t+1}}}\left[V_{t+1}\left(\boldsymbol{r_{t+1}}, \boldsymbol{\omega_{t+1}}\right)\right], \tag{11}$$

$$\text{s.t.} \quad \left(\boldsymbol{x_t}, \boldsymbol{r_{t+1}}\right) \in \mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) \tag{12}$$

Here, $V_t\left(\cdot\right)$ denotes the optimal value-to-go function before setting prices, and $U_t\left(\cdot\right)$ denotes the optimal value-to-go function before dispatching EVs. At the bottom of Figure 2, we illustrate the value-to-go functions and their relations to the MDP states. Given the terminal value $V_{T+1}\left(\cdot\right) \equiv 0$, we have (10) and (11) recursively defined for all $t = T, T-1, \cdots, 1$.

Theoretically, the optimal policy is obtained by solving (10) and (11) backward from $t = T$ to $t = 1$. However, solving for the optimal policy is impractical given the high-dimensional continuous state-action-probability space. In Sections 4 & 5, we introduce a tractable method that produces practical EV fleet decision policies.

4 Solution strategies

In this section, we focus on the optimal dispatching problem, assuming the pricing policy is given. Fixing $\boldsymbol{p_t^D} = P_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right)$ in (11), we are left with the following dynamic program. For all $t = T, T-1, \cdots, 1$,

$$
\begin{aligned}
V_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t}; P^\pi\right) &= \max_{\boldsymbol{x_t}, \boldsymbol{r_{t+1}}} \sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}} p_{lt} x_{let} + \mathbb{E}\left[V_{t+1}\left(\boldsymbol{r_{t+1}}, \boldsymbol{\omega_{t+1}}, \boldsymbol{\theta_{t+1}}; P^\pi\right)\right] && (13)\\
&\text{s.t.} \quad \left(\boldsymbol{x_t}, \boldsymbol{r_{t+1}}\right) \in \mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) && (14)\\
&\qquad \boldsymbol{p_t^D} = P_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right) && (15)
\end{aligned}
$$

The expectation in (13) is taken with respect to $(\boldsymbol{\omega_{t+1}}, \boldsymbol{\theta_{t+1}})$. Here $V_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t}; P^\pi\right)$ denotes the optimal value-to-go function given the pricing policy P^π. Without causing confusion, we will drop the indicator P^π.

We replace the value-to-go term is with tractable value function approximation (VFA) $\bar{V}_{t+1}\left(\boldsymbol{r_{t+1}}\right)$ in (13).

$$
\begin{aligned}
\bar{V}_t\left(\boldsymbol{r_t}\right) &\approx \mathbb{E}_{(\boldsymbol{\omega_t}, \boldsymbol{\theta_t})}\left[\hat{V}_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t}\right)\right], && (16)\\
\hat{V}_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t}\right) &= \max_{\boldsymbol{x_t}, \boldsymbol{r_{t+1}}} \sum_{l \in \mathcal{L}} \sum_{e \in \mathcal{E}} p_{lt} x_{let} + \bar{V}_{t+1}\left(\boldsymbol{r_{t+1}}\right), && (17)\\
&\text{s.t.} \quad \left(\boldsymbol{x_t}, \boldsymbol{r_{t+1}}\right) \in \mathcal{Y}\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{p_t^D}, \boldsymbol{\theta_t}\right) && (18)\\
&\qquad \boldsymbol{p_t^D} = P_t^\pi\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}\right) && (19)
\end{aligned}
$$

Specifically, we start from $t = T$ with $\bar{V}_{T+1}\left(\cdot\right) \equiv 0$ and construct VFAs backward. In each period t, we establish a VFA $\bar{V}_t\left(\boldsymbol{r_t}\right)$ through a sampling phase and a learning phase. In the sampling phase, we sample a set of fleet states and random realizations $(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t})$. For each sampled $(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t})$, we solve (17) to obtain an objective value $\hat{V}_t\left(\boldsymbol{r_t}, \boldsymbol{\omega_t}, \boldsymbol{\theta_t}\right)$. Then, in the learning phase, we construct state-value function approximation $\bar{V}_t\left(\boldsymbol{r_t}\right)$ that fits the sampled state-value pairs. $\bar{V}_t\left(\boldsymbol{r_t}\right)$ approximates the expected value of being in the fleet state $\boldsymbol{r_t}$ at the beginning of period t before knowing the information state, i.e., (16).

The functional structure of VFAs is critical to the solution quality. The more sophisticated the functional structure we have, the higher the approximation quality we can possibly get in (16). However, a complicated $\bar{V}_{t+1}(\boldsymbol{r_{t+1}})$ results in inefficiency or intractability in solving the optimization problem (17). One possible VFA approach is the separable piecewise linear (SPWL) approximations (Godfrey and Powell, 2002), where $\bar{V}_t(\cdot)$ is assumed to be the sum of a group of (concave) piecewise linear scalar functions. For large-scale problems, linear approximations are more practical, e.g., Al-Kanj et al. (2020).

In this article, we apply and develop an input convex neural network (ICNN) VFA method to generate good quality solutions. (We discuss the necessity of using concave VFAs in the supplemental online materials §B.) ICNN has a wider range of functional representability, which potentially produces finer approximations of the target value functions. Neural networks also enables value-based learning. The SPWL method requires learning via sampled gradients, which may not be available in our case. Problem (17) involves state-dependent prices. Therefore, the sample gradient needs additional consideration. According to the envelop theorem and the chain rule, the exact (sub)gradient is

$$\nabla_{\boldsymbol{r_t}} \hat{V}_t = \frac{\partial \hat{V}_t}{\partial \boldsymbol{r_t}} + \frac{\partial \hat{V}_t}{\partial \boldsymbol{p_t^D}} \frac{\partial \boldsymbol{p_t^D}}{\partial \boldsymbol{r_t}}, \tag{20}$$

(assuming continuity in P_t^π). Here, $\frac{\partial \hat{V}_t}{\partial \boldsymbol{r_t}}$ represents the (sub)gradient assuming the pricing $\boldsymbol{p_t^D}$ is fixed, which is the classic dual solution corresponding to constraints (3). $\frac{\partial \hat{V}_t}{\partial \boldsymbol{p_t^D}}$ represents the value sensitivity with respect to the pricing, which can be obtained from the primal solution to (17). However, the availability of $\frac{\partial \boldsymbol{p_t^D}}{\partial \boldsymbol{r_t}}$ varies depending on the structure of P_t^π.

In the rest of the section, we introduce a partial path based VFA algorithm for the fleet management problem. We provide the details about the ICNN architecture and the training method in the supplemental online materials §C & §D, respectively.

A Reinforcement Learning Framework

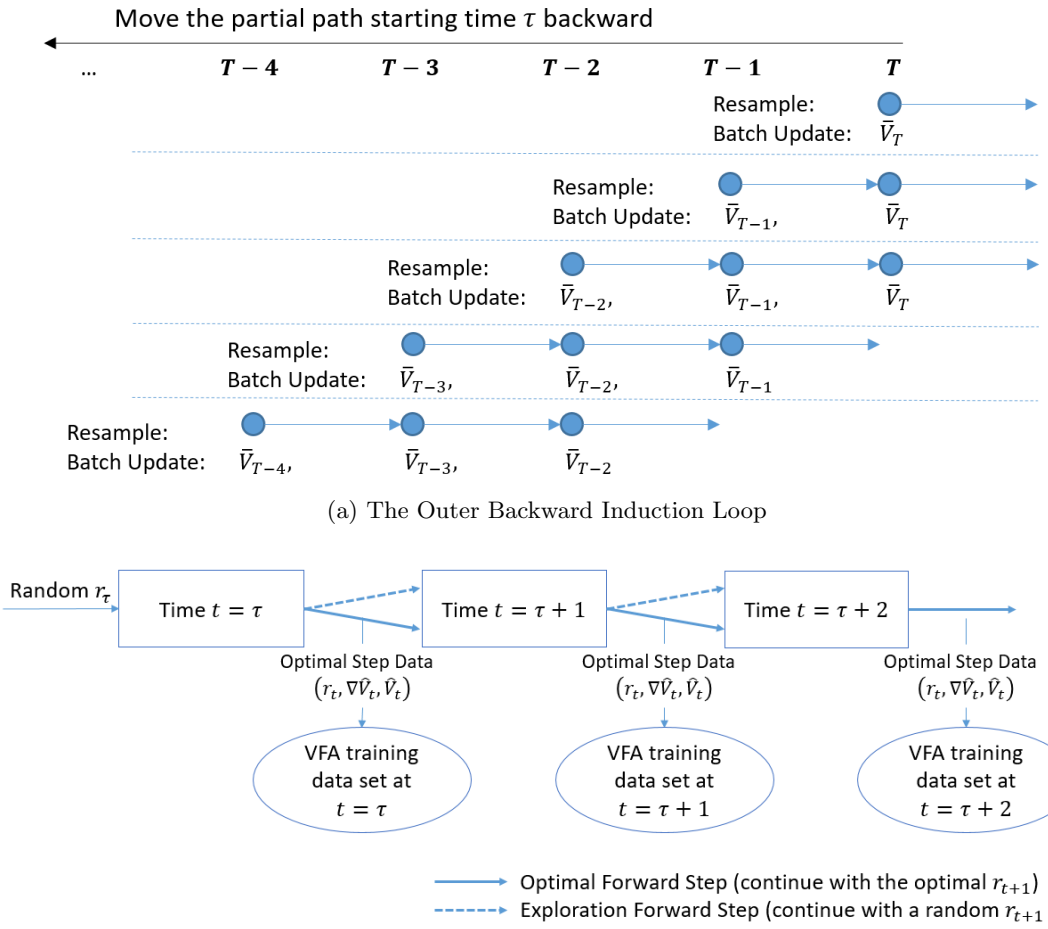
Suppose we have established the subroutines of training VFAs and solving optimization

(17) to generate training data for the previous stage. The algorithm defines a scheme that generates training points and updates VFAs alternately.

We adopt a partial sample path training method. Whenever we finish solving a forward step via (17) from fleet state $\boldsymbol{r_t}$, we obtain one state-gradient-value training data $\left(\boldsymbol{r_t}, \nabla_{\boldsymbol{r_t}} \hat{V}_t, \hat{V}_t\right)$ for the VFA $\bar{V}_t(\cdot)$. Different from using full sample paths, we simulate partial paths from period τ to period $\tau + \Delta\tau$ every time, starting from a uniformly generated initial state r_τ. Repeating the partial paths generation for B times, we collect a batch of B training data for each VFA $\bar{V}_t(\cdot), t = \tau, \tau+1, \cdots, \tau+\Delta\tau$. Then, we update each VFA using the batch. In the outer loop, we move the starting time period τ backward from the last period to the first, so that VFAs are established backward. We call $\Delta\tau$ as the rolling length. Figure 3 is an illustration of the algorithm with a rolling length of two. Algorithm 1 rigorously defines the partial path training method. Algorithm 1 is not limited to the specific choice of VFAs. But for the specific problem and the choice of VFAs, hyperparameters $\Delta\tau$, B and α have to be tuned for better training outcomes.

On one hand, we use partial paths to improve learning efficiency. Before $\bar{V}_{t+1}(\cdot)$ is established with a roughly good shape, the training data at t generated from solving (17) tends to be far from the target value. Nonetheless, solving (17) costs considerable computation time. Therefore, we focus on the critical segments of sample paths and build VFAs through a backward scan.

On the other hand, to obtain good quality VFAs from training points, we need a "smart" spread of visits on the state space. Ideally, $\bar{V}_t(\cdot)$ should capture both the global behavior of $\mathbb{E}\left[\hat{V}_t(\cdot)\right]$ and the local behavior around the states which are frequently visited by the optimal policy. This is achieved via sample path training. Although starting from a uniformly generated initial state, the states visited in the later periods distribute non-uniformly because of optimal forward actions. We also need to retain some visits to the "unlikely" states. This is handled by the exploration probability α in each forward step.



(a) The Outer Backward Induction Loop

(b) The Inner Forward Partial Path

Figure 3 Illustration of Algorithm 1

Subroutines:
StateGenerator(t): uniformly generates a fleet state $\boldsymbol{r}_t$ from the state space
EnvSimulator(t): sample a pair of $(\boldsymbol{\omega}_t, \boldsymbol{\theta}_t)$ from the environment simulator
TrainVFA$\left(\bar{V}_t(\cdot), TrainSet_t\right)$: train the VFA on the training data set
Input:
$\Delta\tau$ = Rolling Time Length
B = number of sample paths in each training iteration
α = Exploration Rate
Initialize VFAs $\bar{V}_t(\cdot), \forall t = 1, 2, \cdots, T$ and $\bar{V}_{T+1}(\cdot) \equiv 0$
for $\tau = T, T-1, \cdots, 1$ **do**
  $FwdPathTimeRange \leftarrow \{\tau, \tau+1, \cdots, \min\{T, \tau+\Delta\tau\}\}$
  $TrainSet_t \leftarrow \emptyset, \forall t \in FwdPathTimeRange$
  for $b = 1, \cdots, B$ **do**
    $\boldsymbol{r}_\tau \leftarrow$ StateGenerator(t)
    for $t \in FwdPathTimeRange$ **do**
      $(\boldsymbol{\omega}_t, \boldsymbol{\theta}_t) \leftarrow$ EnvSimulator(t)
      Solve (23) with $(\boldsymbol{r}_t, \boldsymbol{\omega}_t)$ to obtain the trip pricing $\boldsymbol{p}_t^D$
      Solve (17) with $\left(\boldsymbol{r}_t, \boldsymbol{\omega}_t, \boldsymbol{p}_t^D, \boldsymbol{\theta}_t\right)$ to complete one-period forward step
        $\boldsymbol{r}_{t+1} \leftarrow$ the next state at $\tau+1$ under optimal dispatching
        $grad \leftarrow$ the dual information corresponding to $\nabla_{\boldsymbol{r}_t} \hat{V}_t(\boldsymbol{r}_t, \boldsymbol{\omega}_t, \boldsymbol{\theta}_t)$
        $obj \leftarrow$ the optimal objective value $\hat{V}_t(\boldsymbol{r}_t, \boldsymbol{\omega}_t, \boldsymbol{\theta}_t)$
      add $(\boldsymbol{r}_t, grad, obj)$ to $TrainSet_\tau$
      $\boldsymbol{r}_{t+1} \leftarrow \begin{cases} \boldsymbol{r}_{t+1} & \text{with probability } 1-\alpha \\ \text{a uniformly generated state} & \text{with probability } \alpha \end{cases}$
      continue to the forward simulation at $t+1$ with state $\boldsymbol{r}_{t+1}$
    end
  end
  TrainVFA$\left(\bar{V}_t(\cdot), TrainSet_t\right), \forall t \in FwdPathTimeRange$
end
Output: $\bar{V}_t(\cdot), \forall t = 1, 2, \cdots, T$

Algorithm 1: A reinforcement learning framework

5 Pricing strategy

The VFA techniques cannot be easily extended to (10) when pricing decisions need to be optimized as well. The main reason is that tractable VFA methods (e.g., linear/concave approximation) fail to adequately capture the non-convex shape of the value function $U_t(\cdot)$ (even in low-dimensional, simple cases). Therefore, researchers have turned to other methods to optimize the pricing policy. Topaloglu and Powell (2007) develop a numerical method to find a set of predetermined prices for the EV operating horizon. Their model accounts for the temporal demand imbalance and improves operating profit, but does not react to real-time supply-demand mismatch due to demand uncertainties. Al-Kanj et al. (2020)

introduce a surge-pricing model that decides prices by maximizing the product of price and acceptance rate. The acceptance rate functions are characterized by logistic models and updated by a Bayesian approach. In this article, we use a pricing approach by solving a program embedded in the original EV fleet problem (9).

Our idea is to set prices assuming the demand noise θ_{lt} vanishes. When θ_{lt} degenerates to 0, (10) and (11) reduce to a one-stage problem. For a given fleet state $\boldsymbol{r_t}$ and information state $\boldsymbol{\omega_t}$ and at time t, the optimal pricing and dispatching are obtained by solving the following system.

$$\max_{\boldsymbol{p_t^D}} \max_{(\boldsymbol{x_t},\boldsymbol{r_{t+1}})\in\mathcal{Y}(\boldsymbol{r_t},\boldsymbol{\omega_t},\boldsymbol{p_t^D},\mathbf{0})} \sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}} p_{lt}x_{let} + \mathbb{E}_{\boldsymbol{\omega_{t+1}}}\left[V_{t+1}\left(\boldsymbol{r_{t+1}},\boldsymbol{\omega_{t+1}}\right)\right]. \tag{21}$$

In (21), the pricing and dispatching decisions are made simultaneously. The optimal pricing is always the highest value given the number of cars assigned to the corresponding customer trip. That is,

$$p_{lt} = \frac{1}{b_{lt}}\left(a_{lt} - \sum_{e\in\mathcal{E}} x_{let}\right), \forall l \in \mathcal{L}_D. \tag{22}$$

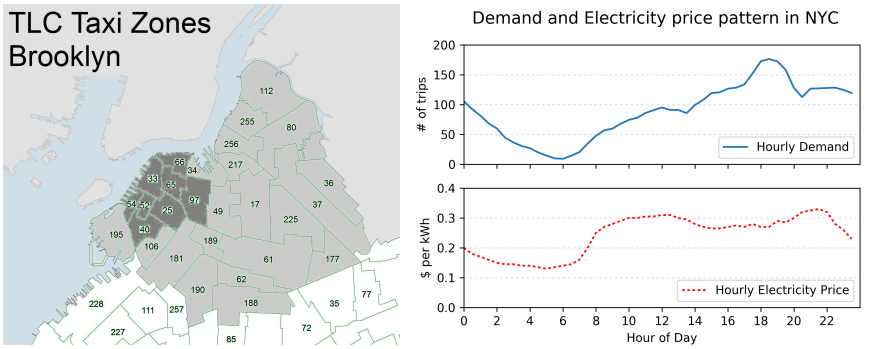
We define $\bar{\mathcal{Y}}\left(\boldsymbol{r_t},\boldsymbol{\omega_t},\boldsymbol{p_t^D}\right)$ as a modified feasible set by replacing (5) with (22). Our pricing strategy is to set

$$P_t^{\pi}\left(\boldsymbol{r_t},\boldsymbol{\omega_t}\right) = \left[\arg\max_{\boldsymbol{x_t},\boldsymbol{r_{t+1}},\boldsymbol{p_t^D}} \sum_{l\in\mathcal{L}}\sum_{e\in\mathcal{E}} p_{lt}x_{let} + \mathbb{E}_{\boldsymbol{\omega_{t+1}}}\left[V_{t+1}\left(\boldsymbol{r_{t+1}},\boldsymbol{\omega_{t+1}}\right)\right]\right]_{\boldsymbol{p_t^D}} \tag{23}$$

$$\text{s.t.} \qquad \left(\boldsymbol{x_t},\boldsymbol{r_{t+1}}\right) \in \bar{\mathcal{Y}}\left(\boldsymbol{r_{t+1}},\boldsymbol{\omega_t},\boldsymbol{p_t^D}\right) \tag{24}$$

The outer bracket $[\cdot]_{\boldsymbol{p_t^D}}$ represents the pricing components of the optimal solution.

The proposed pricing method only relies on the value functions $\mathbb{E}_{\boldsymbol{\omega_t}}\left[V_t\left(\boldsymbol{r_t},\boldsymbol{\omega_t}\right)\right], \forall t \in \mathcal{T}$, which is handled by VFA methods. The pricing policy is automatically updated as we update VFAs in Algorithm 1. This saves us the effort to maintain an independent pricing model. Except for the value-to-go term, (23) is a concave quadratic program if we substitute $\boldsymbol{p_t^D}$ with $\boldsymbol{x_t}$ using (22). Thus, when applying linear/concave VFA to the value-to-go function, (23) remains a concave maximization problem that can be efficiently solved.



(a) Selected zones (b) Hourly demand and electricity price

Figure 4 Input empirical data

6 Numerical experiments

We use New York City TLC data to generate station-to-station demand distribution and perform various numerical experiments on the dynamic EV fleet management problem. We compare the performance of the two VFA methods on a deterministic problem. We apply ICNN method to solve the rest stochastic instances. We show some micro-level characteristics of the policy and sample paths produced by the policy. We also study the interaction between the demand fluctuation and the wholesale electricity price surge using ICNN VFAs.

6.1 *Experiment setup*

All experiments setup are based on the TLC Trip Records (Green Taxi June 2018) in Brooklyn. TLC divides the entire Brooklyn region into 61 zones. Trip records indicate that the majority of the travels occur in the north part of Brooklyn. The grayed 28 zones in Figure 4a account for over 80% of the total trip counts. (We preprocess the zones, merging three adjacent zone pairs: (54,52), (34, 217) and (190, 62) since they have relatively small travel demand.) Assume one EV station is built in each zone. Station capacity is assumed

to be 25 and the total number of EVs operating in the system is 10 times the number of stations. We let each operational period be 30 minutes. This time step setup is assumed according to the taxi data and the estimation from Google Map.

The linear demand curve on each lane is generated from the TLC Trip Records. The solid line in Figure 4b illustrates the average number of trips among these zones every 30 minutes in a day. We assume the price elasticity to be $b_{lt} \equiv 0.125$ per dollar. Then, the average demand potential on lane l at time t is given by $\bar{a}_{lt}$ = the average number of trips+ $b_{lt} \times$ the average trip fare. On the certain lanes where there is no trip occurred in a period, we assume the demand potential $a_{lt} = 1$.

For the EVs in the fleet, we assume their configuration based on the profile of the 2018 Smart Fortwo electric drive (Daimler AG), which is used in car2go car-sharing services. The battery capacity is $\bar{e} = 5$ SoC units (or 15kWh so that 1 SoC unit = 3kWh). The charging rate is 3 SoC units per period (i.e., 9kWh/30min). The energy consumption for driving is 1 SoC unit per trip.

A typical hourly electricity price pattern in NYC is illustrated by the dashed line in Figure 4b. The price on average is about 23 cents per kWh (SAJIP, 2019) and fluctuates between 12 cents and 35 cents per kWh in a day. Thus, we assume a time-varying charging cost $\bar{p}_t^C$ per period, ranging from \$0.6 to \$1.7 per SoC unit. Besides, we assume the rebalance cost is \$5 per period for moving a car between any pair of stations, and the idle cost is 0.

We assume the random information (predicted demand potential, demand noise, and the electricity price) in the model is sampled based on the following parameterized distributions.

$$a_{lt} \sim \text{Gamma Distribution with mean } \bar{a}_{lt} \text{ and CV} = CV_D, \forall l \in \mathcal{L}_D \tag{25}$$

$$\theta_{lt} \sim \text{Normal Distribution with mean 0 and sdv.} = \sigma_\theta \cdot a_{lt}, \forall l \in \mathcal{L}_D \tag{26}$$

$$p_{lt} \sim \text{Normal Distribution with mean } \bar{p}_t^C \text{ and CV} = CV_E, \forall l \in \mathcal{L}_C \tag{27}$$

Table 1 Exogenous model parameters

Model Notation	Meaning	Default Value	Range
$\lvert\mathcal{J}\rvert$	# of EV stations	7	$\{7, 25\}$
$\sum_{j,e} r_{jet}$	Total # of EVs	$10 \times \lvert\mathcal{J}\rvert$	
$\bar{e}$	Maximum SoC level	5	
T	# of periods	48	
$\lvert\varepsilon_l\rvert, l \in \mathcal{L}_C$	Charging Speed	3	$\{1, 2, 3, 4, 5\}$
e^{thr}	Charging threshold	2	$\{0, 1, 2, 3, 4\}$
CV_D	Demand coefficient of variation	0	≥ 0
CV_θ	Demand noise coefficient of variation	0	≥ 0
CV_E	Electricity price coefficient of variation	0	≥ 0

Here, "CV" stands for the coefficient of variation. CV_D, σ_θ, and CV_E control the relative variance of the random variables.

We summarize the exogenous model parameters in Table 1. A specific configuration of the parameters is referred to as a numerical instance. If a parameter is not explicitly declared, it takes the default value indicated in Table 1.

6.2 *Case study results*

In the base case, we examine the deterministic instance. Then, we run stochastic instances with different parameter settings.

We start from a small scale problem to demonstrate the performance of the ICNN method and the intuition of the EV fleet's policy. We consider the travel demand in the dark gray region (in Figure 4a) in a 24-hour planning horizon (i.e., 48 time periods). Let 2:30 a.m. be the beginning period and 2:00 a.m. (next day) be the ending time. We incorporate the low-demand periods from late night to early morning, so that the charging and rebalancing behavior can be better observed. Having the low-demand period at the beginning of the planning horizon also alleviates the influence of truncating the planning horizon, making our 24-hour policy more applicable to the day-by-day long term planning. At the end of the finite planning horizon, the fleet typically reaches a "poor" state (cars

Table 2 Optimality of VFA methods

		SPWL			ICNN			
Initial State	Opt	pred.	simul.	%opt	pred.	%opt	simul.	%opt
random	12448.7	3028.0	12192.8	97.9%	12681.8	101.9%	12361.5	99.3%
±sdv.	±6.5	±12.3	±7.5		±4.7		±8.9	
all 1 SoC	12419.7	2984.9	12166.3	98.0%	12655.4	101.9%	12339.7	99.4%
all 3 SoC	12460.3	3017.3	12206.0	98.0%	12688.1	101.8%	12382.5	99.4%
all 5 SoC	12489.7	3046.3	12236.0	98.0%	12709.3	101.8%	12413.7	99.4%
end-state	12333.3	2602.6	12073.4	97.9%	12607.8	102.2%	12256.8	99.4%

unbalanced and in low-battery). Regardless, they are going to be rebalanced and charged in the low-demand period to prepare for the morning customers.

6.2.1 *VFA optimality gap*

We use a deterministic 7-station experiment to show the performance of the ADP methods. For both the ICNN method and the SPWL method, we train the value functions using Algorithm 1 with fine-tuned hyperparameters until the value functions are stabilized. Table 2 records the $\bar{V}$-predicted profits and actual profits obtained under different methods. The column "Opt" is the optimal profit obtained from solving the extensive formulation of the deterministic dynamic program. A column "pred." denotes the value $\bar{V}_1(\boldsymbol{r_1})$ predicted by the value function at initial state $\boldsymbol{r_1}$. Under the column "simul.", we have the actual profit (i.e., the sum of the profit generated in all 48 periods) produced by the corresponding policy. In the "%opt" columns, we compute their optimality as percentages of the optimal value. We report the predicted and simulated values for different initial states. The first row is the average profit for 50 uniformly generated initial states, with their standard deviations in the next row. The next four rows are initial states where each station contains 10 vehicles with the same SoC. The last column is an "end-state" after a 48-period warm-up phase. This is an exhausted state in which vehicles have no battery left and are ill-distributed.

The results in Table 2 show that ICNN outperforms the SPWL method on both prof-

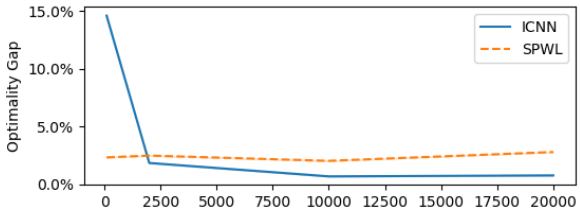


Figure 5 Optimality gap of a deterministic instance

itability and prediction accuracy. Even though the SPWL method produces a good quality solution, the ICNN approximation is able to further slightly improve policy performance. Besides, the approximation value $\bar{V}_t(\cdot)$ is much closer to the actual profit produced under the forward path simulation. In terms of the VFA prediction, the SPWL method approximates the marginal value of state variables, not the total cost, so the value function approximations would not be a prediction of the performance of the policy. The ICNN functions contain predictions close to the simulation result.

We use 10000 training points on each $\bar{V}_t(\cdot)$ for both methods, i.e., $B = 10000$ in Algorithm 1. A one-hidden-layer ICNN value function takes 100 training epochs to reach a minimum training loss, where an epoch corresponds to one scan of the entire training data set. The SPWL method uses two to three epochs to stabilize. In the experiments, the ICNN structure is 42-100-1. The rolling time length and the exploration rate are 3 and 0.2, respectively.

We solve the optimal profit of the deterministic instance to demonstrate the optimality gaps. The gap decreases as shown in Figure 5 when the training batch size increases. The SPWL method requires a small number of training data points to reach maximum performance. With $B = 200$, the SPWL method gets close to a 2% optimality gap. Additional training does not help the SPWL policy to further improve. The ICNN method gets within a 1% gap with $B \geq 5000$. The SPWL method runs for 1.1 hours given a batch size $B = 200$. The ICNN method runs for 8.3 hours given a batch size $B = 5000$. The processor running the experiments is an Intel i5-6500 CPU @ 3.20GHz.

Table 3 Varying the charging speed

Charging Speed	Total Energy Injection	Charging Times	Total Profit
5	1754.7 (+2.54%)	400.4 (-29.8%)	12518.7 (+2.08%)
4	1746.3 (+2.05%)	576.5 (-20.7%)	12456.1 (+1.57%)
3	1711.3 (-0%)	570.2 (-0%)	12263.7 (-0%)
2	1533.1 (-10.41%)	766.5 (+34.4%)	11537.8 (-10.27%)
1	1252.7 (-26.80%)	1252.7 (+119.6%)	9736.4 (-26.37%)

Table 4 Varying the maximum charging threshold e^{thr}

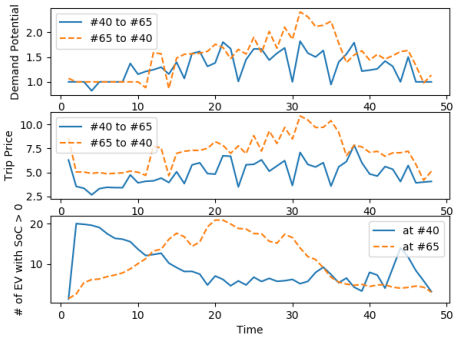
e^{thr}	Total Energy Injection	Charging Times	Avg. Charg. Speed	Total Profit
4	1712.1 (+0.09%)	576.9 (+1.18%)	2.97	12243.1 (+0.01%)
3	1710.9 (+0.02%)	576.5 (+1.11%)	2.97	12242.5 (+0.00%)
2	1710.5 (-0%)	570.2 (-0%)	3	12242.4 (-0%)
1	1693.7 (-0.98%)	564.6 (-0.98%)	3	12192.6 (-0.41%)
0	1680.8 (-1.74%)	560.3 (-1.74%)	3	12152.1 (-0.74%)

6.2.2 *Policy characteristics*

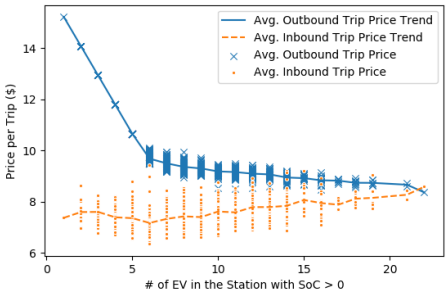
The effects of charging speed and charging threshold

We adjust the charging speed from 1 to 5 SoC per period. (Recall the maximum battery capacity is 5 SoC and the trip energy consumption is 1 SoC per trip.) The results are shown in Table 3, using the average statistics from multiple simulations. It appears that a charging speed of 3 is sufficient to support the fleet operation. The fleet gains a thin profit margin (1%˜2%) by further increasing the charging speed. However, if the charging speed is dropped below 3, the operational performance is more significantly impaired. The fleet loses 26% profit if the charging speed is only 1 SoC per period.

In Table 4, we also display the effect of the charging threshold. Recall that the threshold forbids EVs with SoC above e^{thr} from being charged. The threshold barely affects operational profit. In fact, the VFA policy is "intelligent" enough to prioritize the charging of low-battery EVs (by evaluating the marginal value of SoC). Therefore, charging near-full batteries does not happen often.



(a) Trip price vs OD EV supply



(b) Trip price vs station EV supply

Figure 6 Pricing Strategy Visualization

State-dependent OD pricing

Figure 6a illustrates the supply and pricing variation across the planning horizon for a pair of selected OD (station #40 and #60). The trip pricing mainly follows the trend of customer demand.

The pricing policy presented in Section 5 is also state-dependent. To visualize the effect of supply on the pricing, we pick a specific period and fix the demand at the average level. We show the price and supply relationship of a specific station j in Figure 6b. We uniformly generate $\boldsymbol{r_t}$ from the fleet state space and compute the average of outbound trip prices $\{p_{lt} : l \in \mathcal{L}_D, o_l = j\}$ and the average of inbound trip prices $\{p_{lt} : l \in \mathcal{L}_D, d_l = j\}$. The average prices are plotted against the EV supply in the station, which is quantified as the number of EV with non-zero SoC in station j, i.e., $\sum_{e>0} r_{jet}$. Note that the displayed supply-price relationship has some variation since we aggregate the high-dimensional fleet state into a scalar supply quantity. Figure 6b shows the trend of the average trip price given different supply quantities. If the EV supply in a station is higher/lower, the EV operator tends to decrease/increase the outbound trip prices, and vice versa for the inbound trip prices. A particularly high outbound price occurs when the supply is close to 0. The inbound trip prices also account for their destination supply shortage. A low inbound price

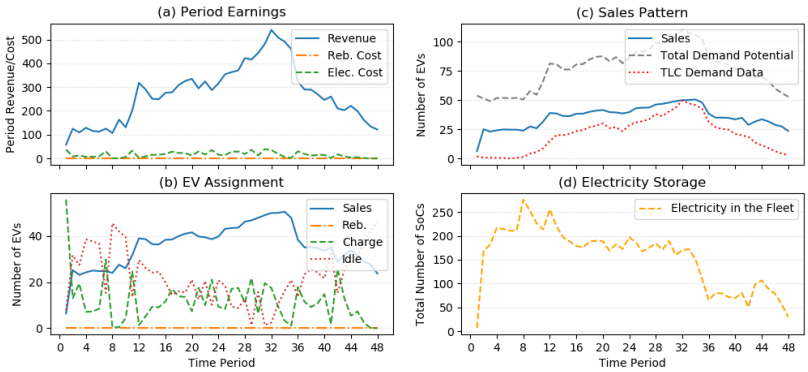


Figure 7 Deterministic forward path

will direct more EVs to station j in the next period.

Sample Path Display

Before preceding to the stochastic model, we show some statistics of the forward path simulation in Figure 7. Figure 7(a) shows the revenue and cost generated in each period. We start from a near-zero-battery end-state. The fleet gradually increases its electricity storage during the first 8 periods (the 4-hour window from 2:00 to 6:00 in the morning) as shown in Figure 7(d). During this time, the fleet mainly gets charged while serving some customer demand. Then, it maintains a charging rate roughly equal to the consumption rate in the middle periods of the day. As the time approaches the end of the planning period, fewer charging tasks are made and the fleet electricity storage drops back to 0. Figures 7(b) and (c) show that the sales and electricity usage remains stable throughout the day, except for the few periods around the period boundaries. With the pricing instrument, the demand is shifted spatial-temporally. The rebalance at a cost of \$5 per trip is never used throughout the day. The EV fleet, with dynamic pricing, consumes electricity steadily. In the next subsection, we show that similar observations hold when the fleet faces random demand.

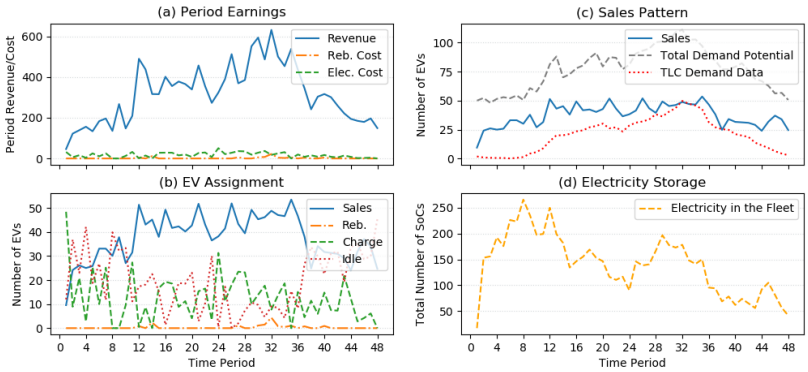


Figure 8 Random sample path ($CV_D = 1$)

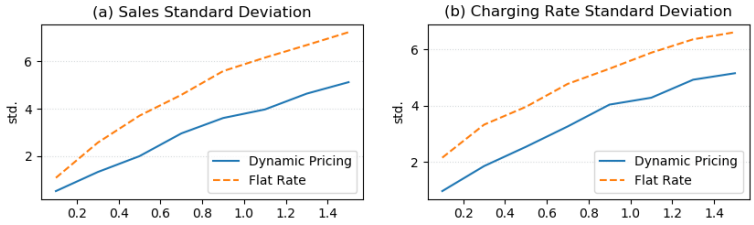


Figure 9 The standard deviation of the number of vehicles assigned to demand serving (left) and charging (right) under different CV_D

6.2.3 *Impact of stochastic demand*

We run instance ($CV_D = 0.3, CV_\theta = 0.5, CV_E = 0.1$) to visualize the fleet's behavior when facing stochastic demand. With uncertain demand potentials (Figure 8(b)), the fleet adjusts its pricing accordingly. This demand-responsive action not only maximizes the profit but also stabilizes the electricity consumption (Figure 8(c)). The electricity storage (Figure 8(d)) exhibits a very similar trend as what we get from the deterministic simulation. Figure 8(a) summarizes the revenue and cost in each period.

To further demonstrate that dynamic pricing stabilizes electricity consumption, we examine the following two statistics when CV_D varies. Let $x_{Dt} := \sum_{l \in \mathcal{L}_D} \sum_{e \in \mathcal{E}} x_{let}$ be the total number of vehicles assigned to demand serving at time t. When facing stochastic demand

$CV_D > 0$, the assignments are uncertain decision variables. We compute the sample standard deviation of x_{Dt} over all sampled demand scenarios, and then take the average over all periods, denoted as $\overline{\text{std}\,[x_{Dt}]}$, which characterizes the uncertainty of the fleet's electricity consumption rate. Similarly, we define $\overline{\text{std}\,[x_{Ct}]}$ as the average sample standard deviation of the number of vehicles assigned charging, standing for the uncertainty of electricity refueling.

Figure 9 shows the trend of these two statistics when CV_D varies from 0.1 to 1.5. As a benchmark, we plot (in dashed lines) the same statistics when a flat rate is used. The solid lines show the uncertainty trend when dynamic pricing is applied. The uncertainty in electricity is significantly reduced when the fleet uses dynamic pricing to regulate demand spatial-temporally. In the simulation of the flat-rate strategy, the fleet also makes assignment decisions anticipating the demand trend. But without the pricing instrument, the fleet operates in a way less responsive to demand surges. The fleet's electricity charging rate is more volatile than that in the dynamic pricing framework.

6.2.4 *Impact of electricity outage*

This part studies the behavior of the fleet when there is a power outage. Although the electricity price could be fluctuating throughout the day, its impact on the fleet's behavior is minor. Comparing to the sales revenue and other major costs, the electricity price is too low to affect the fleet's charging decision. However, this is not to say that the fleet can operate independently without interacting with the power grid. For a large EV fleet, electricity consumption is substantial. Under energy shortage, the grid may limit the fleet's electricity usage (e.g., by setting a charging cap or applying a price surge). Furthermore, a fleet can provide ancillary services to the grid during a power outage. In these scenarios, the fleet will have an operating period with a low or even zero electricity supply. Thus, it makes more sense to study the behavior of the fleet during an electricity outage.

Figure 10 shows a sample path with a 2-hour power outage in the middle of the day.

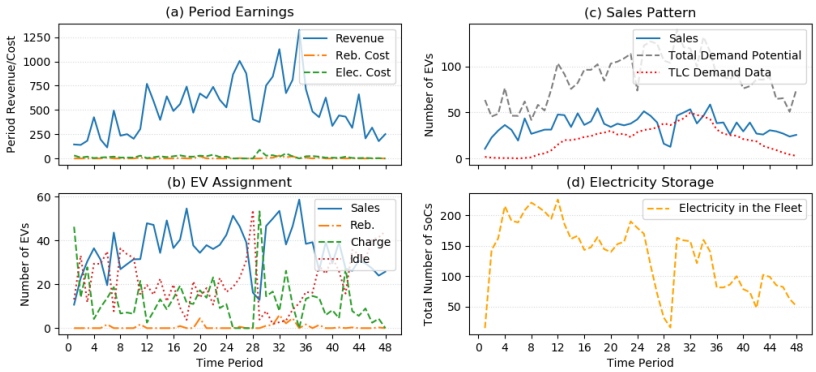


Figure 10 Sample path with a 2-hour power outage (from $t = 25$ to 28)

From period 25 to 28, the EV cannot be charged. After the power is back, we can see an apparent energy replenishment at period 29. The sales begin to drop at period 26 after the outage, and this impact continues to period 29. After period 29, the fleet operates at a normal charging rate and selling rate. In the experiment with $\bar{e} = 3$, the 2-hour outage drains any EV's battery if the vehicle keeps serving demand. However, once there are very few available EVs, the fleet applies higher sales prices to regulate the demand so that the sharing system can keep functioning for longer periods.

6.3 *Insights from the numerical results*

With real demand parameters learned from NYC taxi data, we show that the ICNN based ADP method generates good policy and reasonable value predictions. Then, we apply the method for further study on the behavior of the EV fleet. The EV operator gradually charges and rebalances the fleet in the first 4 hours to prepare for morning demand. With the demand-responsive pricing instrument, very little manual rebalancing is adopted. Moreover, throughout the entire day, EV usage is relatively stable. We also show that the EV fleet significantly reduces the sales and electricity usage variation when customer demand is subject to large uncertainty. The electricity storage inside the fleet is steady

during its main operating period. Even for EVs with small battery capacity, the fleet shows resiliency to electricity power outage.

7 Conclusions

In this article, we presented a dynamic programming model that generates the optimal pricing and EV dispatching decisions for a one-way EV sharing system in a multi-period planning horizon. We showed that, the EV fleet, given the ability set demand-responsive prices for trips over the planning horizon, 1) generates higher operational profit, 2) significantly reduces the need for manual rebalancing, and 3) shows greater resistance to demand fluctuation and stabilize its electricity consumption.

We extended the separable piecewise linear approximation to the input convex neural network (ICNN) approximation. Our method contributes to the dynamic fleet management problem (and other concave dynamic programming problems) in the following aspects. Firstly, ICNNs have greater function representability, and thus better approximate the underlying true value functions. Secondly, ICNN inherits the ability to utilize gradient information in VFA training and further allows the value-based training. The value-based training is helpful when the gradient information cannot be accurately obtained.

This article motivates EV fleet research in various directions. On the modeling side, extending the problem to stochastic multi-period travel time could be interesting but challenging. Future research also can explore the active demand learning of vehicle sharing in dynamic decision processes. On the methodology side, the ICNN method lays the foundation for using non-separable convex VFAs in fleet management problems. It is intriguing to quantify the benefits of non-separable VFAs in solving stochastic dynamic programs.

Acknowledgment

This research is supported by the U.S. National Science Foundation through Grants CNS# 1637772. The authors gratefully acknowledge the valuable comments from the editors and three anonymous reviewers that helped improve the paper.

References

Adler, J. D. and P. B. Mirchandani (2014). Online routing and battery reservations for electric vehicles with swappable batteries. *Transportation Research Part B: Methodological 70*, 285–302.

Al-Kanj, L., J. Nascimento, and W. B. Powell (2020). Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research 284*(3), 1088–1106.

Amos, B., L. Xu, and J. Z. Kolter (2017). Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, Volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155. PMLR.

Boyacı, B., K. G. Zografos, and N. Geroliminis (2017). An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. *Transportation Research Part B: Methodological 95*, 214–237.

Chang, J., M. Yu, S. Shen, and M. Xu (2017). Location design and relocation of a mixed car-sharing fleet with a CO2 emission constraint. *Service Science 9*(3), 205–218.

de Almeida Correia, G. H. and A. P. Antunes (2012). Optimization approach to depot location and trip selection in one-way carsharing systems. *Transportation Research Part E: Logistics and Transportation Review 48*(1), 233–247.

Godfrey, G. A. and W. B. Powell (2002). An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. *Transportation Science 36*(1), 21–39.

Hajibabai, L. and Y. Ouyang (2016). Dynamic snow plow fleet management under uncertain demand and service disruption. *IEEE Transactions on Intelligent Transportation Systems 17*(9), 2574–2582.

He, L., G. Ma, W. Qi, and X. Wang (2021). Charging an electric vehicle-sharing fleet. *Manufacturing & Service Operations Management 23*(2), 471–487.

He, L., H.-Y. Mak, Y. Rong, and Z.-J. M. Shen (2017). Service region design for urban electric vehicle sharing systems. *Manufacturing & Service Operations Management 19*(2), 309–327.

Hodgson, M. J. (1990). A flow-capturing location-allocation model. *Geographical Analysis 22*(3), 270–279.

Lei, C., Z. Jiang, and Y. Ouyang (2019). Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers. *Transportation Research Procedia 38*, 77–97.

Lei, C. and Y. Ouyang (2017). Dynamic pricing and reservation for intelligent urban parking management. *Transportation Research Part C: Emerging Technologies 77*, 226–244.

Li, G. and X.-P. Zhang (2012). Modeling of plug-in hybrid electric vehicle charging demand in probabilistic power flow calculations. *IEEE Transactions on Smart Grid 3*(1), 492–499.

Li, X., J. Ma, J. Cui, A. Ghiasi, and F. Zhou (2016). Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. *Transportation Research Part B: Methodological 88*, 21–45.

Melkote, S. and M. S. Daskin (2001). An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice 35*(6), 515–538.

Nair, R. and E. Miller-Hooks (2011). Fleet management for vehicle sharing operations. *Transportation Science 45*(4), 524–540.

Pan, F., R. Bent, A. Berscheid, and D. Izraelevitz (2010). Locating phev exchange stations in v2g. In *2010 First IEEE International Conference on Smart Grid Communications*, pp. 173–178. IEEE.

Powell, W. B. and H. Topaloglu (2006). Approximate dynamic programming for large-scale resource allocation problems. *INFORMS TutORials in Operations Research*, 123–147.

SAJIP, J. (2019). Overview of electricity rates in nyc. https://www.ny-engineers.com/blog/overview-of-electricity-rates-in-nyc.

Topaloglu, H. and W. Powell (2007). Incorporating pricing decisions into the stochastic dynamic fleet management problem. *Transportation Science 41*(3), 281–301.

Yao, Z., L. H. Lee, W. Jaruphongsa, V. Tan, and C. F. Hui (2010). Multi-source facility location–allocation and inventory problem. *European Journal of Operational Research 207*(2), 750–762.

Zhang, Y., M. Lu, and S. Shen (2021). On the values of vehicle-to-grid electricity selling in electric vehicle sharing. *Manufacturing & Service Operations Management 23*(2), 488–507.

Zipcar (2019). Zipcar overview. https://www.zipcar.com/press/overview.