An approximate dynamic programming algorithm for short-term electric vehicle fleet operation under uncertainty

Lee, Sangmin; Boomsma, Trine Krogh

Published in:
Applied Energy

DOI:
10.1016/j.apenergy.2022.119793

Publication date:
2022

Document version
Peer reviewed version

Document license:
CC BY-NC-ND

Citation for published version (APA):
https://doi.org/10.1016/j.apenergy.2022.119793
An approximate dynamic programming algorithm for short-term electric vehicle fleet operation under uncertainty

Sangmin Lee*
Department of Mathematical Sciences, University of Copenhagen, 2100 København Ø, Denmark
Maersk R&D, 1263 Copenhagen, Denmark sangmin.lee@maersk.com

Trine Krogh Boomsma
Department of Mathematical Sciences, University of Copenhagen, 2100 København Ø, Denmark trine@math.ku.dk

This paper considers the dynamic problem of optimally operating a fleet of plug-in hybrid electric vehicles in a market environment. With uncertainty in future electricity prices and driving demands, we formulate a Markov decision process and determine a cost-minimizing policy for using the engine and charging and discharging the battery. As such, the policy is based on the trade-off between the costs of gasoline and electricity and between current and future power prices. To accommodate an inhomogeneous fleet composition and overcome the computational challenges of stochastic and dynamic optimization, including large-scale state and action spaces, we adopt the methodology of approximate dynamic programming. More specifically, using simulation and value function approximation by linear regression, we apply a least squares Monte Carlo method. This methodology allows for scaling with respect to fleet size and we are able to establish convergence of our algorithm for 100 vehicles by using 5000 samples in the simulation. Our results show that the vehicles should generally discharge the battery rather than using the engine, unless battery capacity is insufficient to fully cover driving demand, but the timing of battery charging should be according to power prices. When comparing our policy to the simple policy of immediate charging, we demonstrate superiority for small and medium-sized fleets, with 2-4% cost differences.

Key words: Plug-in hybrid vehicles; Approximate dynamic programming; Least squares Monte Carlo

1. Introduction

In recent years, electric vehicles (EVs) have attracted considerable attention, and both production and use of EVs continue to widen. Provided that electricity is sourced from renewable energy, such means of transportation directly reduces greenhouse gas emissions and thereby contributes to the green transition.

In general, EVs store energy in their battery packs. For plug-in electric vehicles (PEVs), in particular, the battery pack can be recharged, connecting a plug to an external source of electricity.

* Corresponding author.
By charging in response to price signals from the power market, such vehicles may deliver load balancing, i.e. consume electricity at times of low electricity demand and prices. As such, PEVs can participate in the so-called vehicle-to-grid (V2G) system and thereby support the emergency power system. Readers are referred to [1] for more information on the V2G system. As opposed to gasoline cars, PEVs may have limited driving ranges due to insufficient storage capacity of the battery. For extended driving ranges, plug-in hybrid electric vehicles (PHEVs) are equipped with both an internal combustion engine and an electric motor, and thus, use two different power sources; fuel and electricity.

The present paper takes the perspective of an aggregator that optimally operates a vehicle fleet of PHEVs in a market environment. On behalf of the PHEVs, this aggregator determines a cost-minimizing policy for using the engine and charging and discharging the battery. We focus on short-term operations over time, and thus, consider a dynamic problem. With uncertainty in future electricity market prices and driving demand, we resort to stochastic optimization and formulate a Markov decision process (MDP). In particular, our decisions must adapt to electricity prices and driving demand, which evolve stochastically over time according to some Markov process. Thus, at each point in time, given the state of current battery levels, driving demands and the electricity price, the fleet aggregator makes charging and discharging decisions such as to minimize current costs and expected future costs described by a value function.

We describe the uncertainty in driving patterns, assuming independence between drivers. For each driver, our model relies on a number of simple yet realistic assumptions; the driver makes a trip to work in the morning and back home in the evening and, with a given probability, makes another trip in the afternoon. Departure time, driving distance, and average driving speed of each trip follow given distributions. To model uncertainty in electricity market prices, we assume a first-order autoregressive process, fitted to real historical electricity spot prices.

To capture short-term dynamics of the battery levels, the driving demands and the electricity price with a fine discretization of time, a discrete representation of uncertainty and a considerable fleet of different vehicles, the number of states and actions becomes very large. In order to overcome such computational challenges of stochastic and dynamic optimization, we adopt the methodology of approximate dynamic programming (ADP), and more specifically, the least squares Monte Carlo (LSMC) method [2]. The ADP combines dynamic programming (DP) methods with simulation and function approximation to avoid the so-called curse-of-dimensionality with respect to the sizes of state and action spaces. Based on backward DP, LSMC approximates the value function using linear regression, which is rather computationally inexpensive and facilitates optimization by linear programming.

We summarize our contributions as follows:
• We develop an optimization framework for electric vehicle fleet operation that allows for an
inhomogeneous fleet composition and accounts for time dependencies in decision-making and the
stochastic development in driving patterns and electricity prices over time.
• We adopt an approximate dynamic programming approach and demonstrate how it is able to
handle various dimensions in electric vehicle fleet operation, such as short-term dynamics with
a fine discretization of time, a significant number of different vehicles and multiple sources of
uncertainty.
• We explore the potential of the methodology for a case study of optimal fleet operation with a
range of real-life PHEV models, assessing the quality of the optimal policy as well as the convergence
and scalability properties of our approach.

The remainder of this paper is organized as follows. Section 2 provides a literature review. Section 3 describes the problem of optimal charging and discharging of PHEVs, whereas Section 4 formulates the problem as an MDP using DP. Section 5 presents the algorithm of the LSMC method. Section 6 presents the case study, whereas Section 7 contains numerical experiments. Section 8 summarizes the results and discusses future work.

2. Literature review

A vast amount of literature has developed optimization models for optimal charging of electric
to a public charging station that co-optimizes charging and other distributed renewable energy
sources, with the aim of providing grid services. The authors account for uncertainty in charging
demand, additional electricity load, renewable supply and prices. In contrast to our formulation,
however, their model is limited to two stages of decision-making. Along similar lines, [4] studies an
energy hub of renewable energy sources and electric vehicles with uncertainty on both the demand
and supply sides. In particular, uncertainty involves random departure times, driving distances
and arrival times of the vehicles. Optimization is restricted to being scenario-based, whereas each
scenario accounts for different PHEV charging modes such as vehicle-to-grid and grid-to-vehicle.
In addition to battery charging, [5] optimizes vehicle speeds. Further relevant modeling details
include the adaptation of the energy management to the lifespan of the batteries and the impact
of battery aging on the energy consumption, as suggested in [6].

The problem of optimal charging is challenged by the presence of uncertainties. Approaches
to optimization under uncertainty include the mathematical programming methodology, such as
multi-stage stochastic programming, see [3] and [7]. This methodology, however, suffers from poor scalability, with computational complexity increasing dramatically with the number of stages and the number of random variables. For instance, a multi-stage programming problem with \( T \) stages
and $N$ realization of the random variables essentially consists of $N^T$ copies of the deterministic problem, corresponding to the paths through the so-called scenario tree.

Another strand of literature formulates the optimal charging problem as a Markov decision process (MDP) [8] and applies dynamic programming (DP) [9]. The MDP formulation decomposes the problem into successive sub-problems that are much less computationally demanding. As such, an MDP may handle uncertainty more efficiently by solving only $T$ two-stage stochastic problems, each with $N$ scenarios. For example, [10] develops an MDP and applies the DP method to the optimal charging and discharging problem of an aggregator, assuming a Markov chain model for electricity prices (see also [11] for a deterministic formulation). Similarly, [12] proposes an inhomogeneous hidden Markov chain model for driving patterns and applies DP.

The DP method may, however, likewise suffer from scalability problems due to the curse-of-dimensionality with regard to state and action spaces, [13]. While some references manage such computational issues by discretization and aggregation of the state and action spaces, see [10], [12], [14], [15], many references employ variants of approximate dynamic programming (ADP) and benefit from simulation and function approximation. Considering parking-lot EV charging, [16] develops a two-stage ADP algorithm based on a K-nearest neighbor approach and forecasting models for electricity prices and charging demands. This paper takes the perspective of the charging station rather than the vehicle fleet. Optimal charging is determined by the trade-off between current and future electricity prices. In contrast, we model both the battery and the engine of each vehicle such that optimization is also determined by the trade-off between costs of gasoline and electricity. For a station with multiple types of chargers, [17] introduces an energy management system that combines the ADP approach with an evolutionary algorithm suggested in [18]. The authors likewise take the perspective of the charging station and optimize the timing of charging. On the contrary, we account for battery dynamics and determine both the timing and the magnitude of charging. Using the terminology of the machine learning community, in which ADP is also referred to as reinforcement learning, [19] proposes to approximate the value function using a deep neural network and forecasts the price using a long-short-term memory network. This paper focus on a single vehicle, whereas we extend the ADP methodology to a fleet of vehicles.

To handle the large-scale optimal charging problem, the present paper invokes the least squares Monte Carlo (LSMC) method [2]. The LSMC method, pioneered by [20], [2] and [21], is a state-of-the-art approximate dynamic programming methodology that has found extensive use in financial engineering and valuation of options [22]. Applications of this method in other fields include renewable energy investments [23], fleet replacements [24], and stochastic storage problems [25], see also [22] for more applications. To the best of our knowledge, this method have not previously been applied to the specific problem of optimal charging.
3. Electric vehicle operation

<table>
<thead>
<tr>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>EV</td>
</tr>
<tr>
<td>PEV</td>
</tr>
<tr>
<td>PHEV</td>
</tr>
<tr>
<td>V2G</td>
</tr>
<tr>
<td>SOC</td>
</tr>
<tr>
<td>CD</td>
</tr>
<tr>
<td>CS</td>
</tr>
<tr>
<td>MPG</td>
</tr>
<tr>
<td>MPGe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J )</td>
</tr>
<tr>
<td>( {0, \ldots, T} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{\text{max}} )</td>
</tr>
<tr>
<td>( l_{\text{max}}^j )</td>
</tr>
<tr>
<td>( l_{\text{min}}^j )</td>
</tr>
<tr>
<td>( p_t )</td>
</tr>
<tr>
<td>( \bar{p} )</td>
</tr>
<tr>
<td>( \alpha )</td>
</tr>
<tr>
<td>( \gamma_{\text{cs}}^j )</td>
</tr>
<tr>
<td>( \gamma_{\text{cd}}^j )</td>
</tr>
<tr>
<td>( d_{j,t} )</td>
</tr>
<tr>
<td>( \eta_{\text{cd}}^j )</td>
</tr>
<tr>
<td>( \eta_{\text{cs}}^j )</td>
</tr>
<tr>
<td>( \eta_{\text{cr}}^j )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_{j,t} )</td>
</tr>
<tr>
<td>( v_t )</td>
</tr>
<tr>
<td>( \hat{p}_t )</td>
</tr>
</tbody>
</table>

**Table 1** Nomenclature.

We consider the short-term operation of a fleet of PHEVs. A PHEV consumes both electricity and fuel, discharging a battery and using an internal combustion engine, respectively. As the name suggests, the battery is charged by plugging a cable into an external power source, referred to as a charging station. Each vehicle serves the demand of its driver. A so-called aggregator is responsible for the management and operation of the fleet. We assume that the aggregator has full control of the vehicles. This may be by owning a fleet of trucks or a car-sharing service, or it may be that vehicle owners enter into agreements with an aggregator. The drivers need to contact the
Figure 1  Interactions between the EV fleet, the EV aggregator, the DSO/TSO and the power market.

aggregator’s control center to receive a charging schedule or this schedule needs to be conveyed via electronic signals to the driver’s charger. As illustrated in Figure 1, the aggregator interacts with the distribution system operator (DSO) and/or the transmission system operator (TSO), who are responsible for the distribution of energy to the consumers and manage the transmission from producers to consumers through the grid. Most importantly, the aggregator acts as a wholesale agent in the power market. With a sufficiently large volume, the aggregator can trade in the electricity market and purchase electricity on behalf of the fleet. This allows drivers to benefit from adapting their charging schedule to the development in market prices, i.e. to consume electricity at times of low prices.

Here, we take the perspective of the aggregator optimizing the operating of its fleet. Decision-making involves the volumes and timing of electricity purchases in response to variations in electricity prices and driving demand, assuming that discharging and use of the engine of each vehicle is likewise determined by optimization.

For the modeling of short-term operation, we assume a finite and discrete time horizon defined by the time points \( \{0, \ldots, T\} \). We let \( t \) refer to a specific point in time and consider the corresponding time interval \([t, t + 1]\) (the interval \([t, t + 1]\) = \(\{s : t \leq s < t + 1\}\) contains \(t\) but does not contain \(t + 1\)). In the present paper, we assume that \( t \) represents a number of hours of the time horizon, with the current time being \( t = 0 \) and \( T \) representing the number of hours in a week.

We denote the fleet of PHEVs by \( J = \{0, \ldots, J\} \) and let \( j \) refer to a single vehicle. We model driving demand as exogenous. When the driving demand of a vehicle is zero, we assume that it is parked and plugged in at a charging station, facilitating replenishment of the battery. Our model, however, immediately extends to vehicles that are not always plugged in when parked. We further assume that a vehicle is either parked or driven during an entire time interval. When driven, it can be in one of two modes, depending on the battery state of charge (SOC): charge-depletion (CD) mode, discharging the battery, or charge-sustaining (CS) mode, using the engine. If the battery SOC reaches its minimum level, the vehicle can only be in CS mode. When the battery SOC is
above its minimum level, however, we determine the distribution of driving demand between the two modes by optimization. Note that we do not allow for charging of the battery by using the engine.

Vehicle $j$ is characterized by the following features: its maximum and minimum SOC of the battery, $l_j^{\text{max}}$ and $l_j^{\text{min}}$ (kW), respectively; its energy and fuel efficiency in CD and CS mode, $\eta_j^{\text{cd}}$ (kWh/km) and $\eta_j^{\text{cs}}$ (km/L), respectively; its charging efficiency, $\eta_j^{\text{cr}}$ (%); and its battery depletion cost, denoted by $\gamma_j^{\text{cd}}$ (EUR/kWh). In addition to the vehicle characteristics, we introduce the following parameters: the cost of fuel, $\gamma_j^{\text{cs}}$ (EUR/L), the maximum rate of charge at the station, $v^{\text{max}}$ (kW), and the driving demand in the time interval $[t, t+1]$, denoted $d_{j,t}$ (km).

The model optimizes charging, discharging of the battery and use of the engine by letting the battery level at time $t$, $l_{j,t}$ (kWh), be a decision variable (Note that a battery operated at a rate of energy of $l_j^{\text{min}} \leq l \leq l_j^{\text{max}}$ kW for an hour has potential to generate $l$ kWh of energy). However, the current battery level, $l_j$, (kWh), is an exogenous parameter to the model. For ease of notation, we define the vectors $l_t = (l_{j,t}, \ldots, l_{J,t}) \in \mathbb{R}^J$ and $d_t = (d_{j,t}, \ldots, d_{J,t}) \in \mathbb{R}^J$.

We assume that the fleet aggregator trades in the electricity market such that charging costs relate directly to market prices. Our model allows for endogenous prices, letting the electricity price in time interval $[t, t+1]$, $\hat{p}_t$ (EUR/kWh), be a decision variable. For simplicity, we consider an affine function $\hat{p}_t = p_t + \alpha \cdot v_t$, where $v_t$ (kWh) is total load from the vehicle fleet, the parameter $p_t$ (EUR/kWh) is the electricity price when total load is zero, and $\alpha$ (EUR/kWh$^2$) is the sensitivity of the price with respect to load. The expected future electricity price at time $T$ is denoted by $\bar{p}_t$.

Both driving demand and electricity prices may be random. To capture this, we let $\{d_t : t = 0, 1, \ldots, T-1\}$ and $\{p_t : t = 0, 1, \ldots, T-1\}$ be stochastic processes of random variables that realize over time. We assume that these processes are Markovian, i.e. that $d_{t+1}, p_{t+1} \mid d_t, p_t$ has the same distribution as $d_{t+1}, p_{t+1} \mid d_t, p_t$, where $\cdot \mid \cdot$ refers to conditioning. We further assume that neither $d_t$ or $p_t$, are affected by our decisions. The battery levels, however, must be adapted such that $l_t$ only depends on the information available at time $t$, i.e., $d_t$ and $p_t$.

The notation is summarized in Table 1.

4. Dynamic programming

We formulate the optimization problem for fleet operation, using the dynamic programming terminology of [13], see also Table 2 for the notation.

---

1 We assume that it is clear from the context whether a vector is a row or column vector and leave out the use of the transpose.
Dynamic programming terminology

- $S$: set of available states at time $t$
- $S_t$: (pre-decision) state at time $t$
- $S_a^t$: post-decision state at time $t$
- $A$: set of available actions
- $a_t$: decision/action at time $t$
- $f_t^s(S_t, a_t)$: value of transition function, given action $a_t$ is made in state $S_t$
- $C_t(S_t, a_t)$: value of cost function, given action $a_t$ is made in state $S_t$
- $V_t(S_t)$: value of (pre-decision) state $S_t$ at time $t$
- $V^a_t(S^a_t)$: value of post-decision state $S^a_t$ at time $t$
- $A^\pi_t$: decision rule at time $t$ under policy $\pi$
- $\Pi$: set of admissible policies

Table 2 Nomenclature.

For $t = 0, 1, \ldots, T$, we denote the state of the fleet by $S_t$. The state variable captures information on the current battery levels, the driving demand of the vehicles and the electricity price, except at time $T$ it depends on the future price of electricity:

$$S_t = \begin{cases} (l_t, d_t, p_t), & \text{for } t = 0, 1, \ldots, T - 1, \\ (l_t, \bar{p}), & \text{for } t = T. \end{cases}$$

The state consists of an endogenous component, $l_t$, determined by our decisions, and two exogenous components, $d_t$ and $p_t$, evolving according to specific stochastic processes independently of our decisions. We denote the set of available states at time $t$ by $S = [l^\text{min}_1, l^\text{max}_1] \times \cdots \times [l^\text{min}_J, l^\text{max}_J] \times \mathbb{R}_+^J \times \mathbb{R}_+ \subseteq \mathbb{R}^{2J+1}$.

At time $t = 0, 1, \ldots, T - 1$, the decision-maker executes an action $a_t = (a_{1,t}, \ldots, a_{J,t})$, representing the change in battery levels (kWh) during the interval $[t, t + 1]$. When not driving, a vehicle is plugged in and the battery can be replenished. Thus, the change in battery level is non-negative and bounded above by the maximum rate of charge at the station and the idle battery capacity. When driving, however, the change in battery level is non-positive and bounded below by the amount of energy required to serve driving demand and the remaining battery capacity. We implicitly assume that the capacity of the engine can always accommodate the driving demand of a time interval. On this basis, we define the state-dependent set of actions as follows:

$$\mathcal{A}_j(S_t) = \begin{cases} \{a_{j,t} : 0 \leq a_{j,t} \leq \min\{l^\text{max}_{j,t}, v^\text{max}\}\}, & \text{if } d_{j,t} = 0, \\ \{a_{j,t} : -\min\{l_{j,t} - l^\text{min}_{j,t}, \eta_{j,t} \cdot d_{j,t}\} \leq a_{j,t} \leq 0\}, & \text{if } d_{j,t} > 0, \end{cases}$$

with $\mathcal{A}(S_t) = \mathcal{A}_1(S_t) \times \cdots \times \mathcal{A}_J(S_t)$ and the set of available actions denoted by $\mathcal{A} = \cap_{S_t \in S} \mathcal{A}(S_t)$. 
A state transition is defined by the function $f_t : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, taking the current state $S_t$ and action $a_t$ as input and outputting the next state $S_{t+1}$:

$$S_{t+1} = f_t(S_t, a_t) = (l_t + a_t, d_{t+1}, p_{t+1}).$$

The execution of action $a_t$ in state $S_t$ incurs a cost. When not driving, the unit cost depends on the electricity price and the efficiency of charging. When driving, the unit cost relates to battery depletion or fuel consumption, taking into account the energy and fuel efficiencies, respectively. Note that battery depletion translates into a change in battery level, whereas any remaining driving demand must be covered by using the engine. For $t = 0, 1, \ldots, T - 1$, the cost function $C_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as follows:

$$C_t(S_t, a_t) = \sum_{j \in J} C_{j,t}(S_t, a_t),$$

where

$$C_{j,t}(S_t, a_t) = \begin{cases} \hat{p}_t \cdot a_{j,t} \cdot \frac{1}{\eta_j}, & \text{if } d_{j,t} = 0, \\ \gamma_{j}^{cd}(-a_{j,t}) + \gamma_{j}^{cs} \cdot \frac{1}{\eta_j} (d_{j,t} + a_{j,t} \cdot \frac{1}{\eta_j}), & \text{if } d_{j,t} > 0, \end{cases}$$

with

$$\hat{p}_t = p_t + \alpha \times v_t, \quad v_t = \sum_{j : d_{j,t} = 0} a_{j,t}.$$  

At time $T$, we let the cost function reflect the future costs of charging:

$$C_T(S_T) = \bar{p} \sum_{j \in J} (l_{j,0} - l_{j,T}) \cdot \frac{1}{\eta_j^T}.$$  

With these specifications, the problem of operating the fleet, while considering random driving demands and electricity prices, can be formulated as a stochastic optimization problem. We refer to a policy $\pi$ as a sequence of decision rules, i.e., $\pi = \{A_0^\pi, A_1^\pi, \ldots, A_{T-1}^\pi\}$, where $A_t^\pi : \mathcal{S} \to \mathcal{A}$ is a function that takes a state $S_t \in \mathcal{S}$ as input and outputs an action $a_t \in \mathcal{A}$ according to policy $\pi$. Moreover, we let $\Pi$ be the set of all admissible policies, i.e., all $\pi$ such that $S_t \in \mathcal{S}, a_t \in \mathcal{A}(S_t)$ and $S_{t+1} = f_t(S_t, a_t)$ for all $t = 0, 1, \ldots, T - 1$ and $S_T \in \mathcal{S}$. The problem is now:

$$\min_{\pi \in \Pi} \mathbb{E} \left[ \sum_{t=0}^{T-1} C_t(S_t, A_t^\pi(S_t)) + C_T(S_T) \ \middle| \ S_0 \right],$$

where $\mathbb{E}[\cdot | \cdot]$ refers to a conditional expectation.

For the application of dynamic programming, we let $V_t(S_t)$ be the value of being in state $S_t$. By the principle of optimality, the value function satisfies the recursion

$$V_t(S_t) = \min_{a_t \in \mathcal{A}(S_t)} \left\{ C_t(S_t, a_t) + \mathbb{E}[V_{t+1}(S_{t+1}) | S_t, a_t] \right\},$$

where
for \( t = 0, 1, \ldots, T - 1 \). At time \( T \), the value is given by

\[
V_T(S_T) = C_T(S_T). \tag{3}
\]

This decomposes the problem (1) into \( T \) stages that can be solved sequentially.

The recursive equations (2) and (3) use so-called pre-decision state variable, representing the state of the fleet prior to making decisions. Alternatively, we may define a recursion in terms of post-decision states and reformulate the problem using post-decision state variables as follows.

For \( t = 0, 1, \ldots, T - 1 \), we denote the post-decision state variable by \( S^a_t \). This represents the state of the fleet upon making the decision \( a_t \) but prior to the arrival of new information \( d_{t+1}, p_{t+1} \). The post-decision state is therefore given by the next battery levels, the current driving demands of the vehicles and the electricity price:

\[
S^a_t = (l_{t+1}, d_t, p_t), \quad \text{for } t = 0, 1, \ldots, T - 1.
\]

The state transition is defined by the function \( f^a_t : S \times A \rightarrow S \), specifying the post-decision state \( S^a_t \) as a function of the pre-decision state \( S_t \) and action \( a_t \):

\[
S^a_t = f^a_t(S_t, a_t) = (l_t + a_t, d_t, p_t).
\]

The post-decision value function \( V^a_t \) is then

\[
V^a_t(S^a_t) = \mathbb{E}[V^a_{t+1}(S^a_{t+1}) \mid S^a_t], \quad \text{for } t = 0, \ldots, T - 1. \tag{4}
\]

Using (4), we have from (2) that

\[
V_t(S_t) = \min_{a_t \in A_t(S_t)} \left\{ C_t(S_t, a_t) + V^a_t(f_t(S_t, a_t)) \right\}, \tag{5}
\]

for \( t = 0, \ldots, T - 1 \). If (5) is plugged into (4), we obtain the recursion

\[
V^a_t(S^a_t) = \mathbb{E} \left[ \min_{a_{t+1} \in A_{t+1}(S^a_t)} \left\{ C_{t+1}(S_{t+1}, a_{t+1}) + V^a_{t+1}(f^a_t(S_{t+1}, a_{t+1})) \right\} \mid S^a_t \right], \tag{6}
\]

for \( t = 0, \ldots, T - 2 \). When \( t = T - 1 \), the value is

\[
V^a_{T-1}(S^a_{T-1}) = \mathbb{E} [C_T(S_T) \mid S^a_{T-1}]. \tag{7}
\]

For further details of dynamic programming with the post-decision states, we refer readers to Chapter 4 of [13].

We have described two ways of solving the stochastic optimization problem (1): solving (2)-(3) using the pre-decision value functions \( V_t(\cdot) \), and solving (6)-(7) using the post-decision value functions \( V^a_t(\cdot) \). With the pre-decision state variable, the minimization problem (2) involves the expectation of the value function and, thus, is stochastic. With the post-decision state variable, however, the minimization problem in (6) does not involve an expectation but is a deterministic optimization problem. Throughout the rest of the paper, we apply the post-decision formulation.
5. Approximate dynamic programming

Solving the recursive equations (6) and (7) can quickly become intractable for larger state spaces due to the curse-of-dimensionality. For our problem, the state space increases dramatically with the number of vehicles in the fleet, \( J \) (if the battery levels are discretized, the state space increases exponentially with \( J \)). Approximate dynamic programming overcomes this issue by using simulation and value function approximation. Here, we apply the least squares Monte Carlo method, which employs Monte Carlo simulations and linear regression.

We denote by \( \overline{V}_i^a(S_i^n) \) an approximation of the post-decision value function \( V_i^a(S_i^n) \) and adopt a linear function approximation as follows. We let \( X_i = (1, x_{1,t}, x_{2,t}, \ldots, x_{K,t}) \) be a \((K+1)\)-dimensional vector corresponding to the post-decision state \( S_i^n = (l_{t+1}, d_t, p_t) \) such that \( K = 2 \cdot J + 1 \) and

\[
x_{1,t} = l_{1,t+1}, \quad x_{2,t} = l_{2,t+1}, \ldots, \quad x_{J+1,t} = d_{J+1,t+1}, \quad x_{J+1,t} = d_{J+2,t}, \ldots, \quad x_{K+2,t} = d_{K+2,t}, \quad x_{K+2,t} = p_t.
\]

The linearity of the post-decision value function approximation implies that

\[
\overline{V}_i^a(S_i^n) = X_i \beta_t = \beta_0 \cdot x_{1,t} + \beta_1 \cdot x_{2,t} + \beta_2 \cdot x_{J+2,t} + \cdots + \beta_K \cdot x_{K+2,t},
\]

where \( \beta_t = (\beta_0, \beta_1, \ldots, \beta_K) \) is a vector of parameters.

---

**Algorithm 1**: LSMC for electric vehicle operation

**Step 0** Generate \( N \) sample paths of state information, \((l^n_0, \ldots, l^n_T, d^n_0, \ldots, d^n_J, p^n_0, \ldots, p^n_T, \bar{p}^n)\), \( n = 1, 2, \ldots, N \).

**Step 1** For \( t = T - 1, T - 2, \ldots, 0 \):

**Step 1.1** For each sample path, determine \( \hat{y}_t^n \) as follows:

if \( t = T - 1 \),

\[
\hat{y}_t^n = C_{t+1}(S_{t+1}^n),
\]

if \( t = T - 2, T - 3, \ldots, 0 \),

\[
\hat{y}_t^n = \min_{a_{t+1} \in A_{t+1}(S_{t+1}^n)} \left\{ C_{t+1}(S_{t+1}^n, a_{t+1}) + \overline{V}_{t+1}^a(S_{t+1}^n, a_{t+1}) \right\}.
\]

**Step 1.2** Carry out a two-norm linear regression on \( \{(X_t^n, \hat{y}_t^n) : n = 1, \ldots, N\} \), where

\[
X_t^n = (1, l^n_{1,t+1}, l^n_{2,t+1}, \ldots, l^n_{J,t+1}, d^n_{1,t}, d^n_{2,t}, \ldots, d^n_{J,t}, p^n_t)
\]

and determine the least squares estimator \( \hat{\beta}_t \).

**Step 1.3** Let \( \overline{V}_i^a(S^n) = X_i \hat{\beta}_t \), where

\[
X_t = (1, l^n_{1,t+1}, l^n_{2,t+1}, \ldots, l^n_{J,t+1}, d^n_{1,t}, d^n_{2,t}, \ldots, d^n_{J,t}, p^n_t) = (1, S^n).
\]

**Step 2** Return \( \hat{\beta}_t, t = 0, 1, \ldots, T - 1 \).
Algorithm 1 summarizes the LSMC method. We generate sample paths of state information and iterate backward in time. At each stage, we compute the value estimate for each sample, using the value function approximation obtained in the previous stage, and we determine the current approximation by linear regression.

To generate sample paths for the state information, we use conditional sampling from the exogenous stochastic processes of electricity prices and driving patterns. For the endogenous battery levels, we rely on Latin hypercube sampling (LHS) as suggested by [26] for high-dimensional state spaces. LHS divides the space of battery levels into so-called Latin hypercubes such that a sample is the only one in its dimension. More specifically, the battery level of a vehicle must be in one of the intervals \([l_{\text{min},h}, l_{\text{max},h}], h = 1, \ldots, H\). If, in the \(n\)-th sample, the battery level of vehicle \(j\) is in the \(h\)-th interval, \(l_{j,t}^n \in [l_{\text{min},h}, l_{\text{max},h}]\), then \(l_{j,t}^n \notin [l_{\text{min},h'}, l_{\text{max},h'}]\) for \(h' \neq h\) and \(l_{j',t}^n \notin [l_{\text{min},h}, l_{\text{max},h}]\) for \(j' \neq j\). The LHS method ensures that more dimensions does not involve more samples.

In Step 1.1, the algorithm considers the \(n\)-th sample path and estimates \(V_{t+1}^n(S_{t+1}^n, a_{t+1})\) in (6) with the estimate \(V_{t+1}^n(f_t^n(S_{t+1}^n, a_{t+1}))\) and approximating the expectation
\[
\mathbb{E}\left[\min_{a_{t+1} \in A_t(S_{t+1}^n)} \left\{ C_{t+1}(S_{t+1}^n, a_{t+1}) + \bar{V}_{t+1}^n(f_t^n(S_{t+1}^n, a_{t+1})) \right\} \mid S_t^n \right]
\]
by the sample \(\hat{Y}_t^n\). We take these \(N\) estimates as data. Note that, with \(\bar{V}_{t+1}^n\) being a linear function, the minimization problem that determines \(\hat{Y}_t^n\) is simply a linear programming problem. In Step 1.2, we are given the data set \(\{(X_t^n, \hat{Y}_t^n) : n = 1, \ldots, N\}\) and determine the least squares estimator \(\hat{\beta}_t\) as the minimizer of the squared two-norm, i.e.,
\[
\hat{\beta}_t = \arg\min_{\beta_t} \sum_{n=1}^{N} (\hat{Y}_t^n - X_t^n \beta_t)^2.
\]
This estimate serves as the parameter vector of the post-decision value function approximation \(\bar{V}_t^n(S_t^n)\), see Step 1.3.

Given \(\bar{V}_t^n(S_t^n)\), \(t = 0, \ldots, T - 1\), the optimal action in stage \(S_t\), denoted by \(\hat{a}_t\), is estimated by:
\[
\hat{a}_t = \arg\min_{a_t \in A_t(S_t)} \left\{ C_t(S_t, a_t) + \bar{V}_t^n(f_t^n(S_t, a_t)) \right\}, \quad \text{for } t = 0, 1, \ldots, T - 1.
\]

6. Case study

To demonstrate the applicability and potential of our approach, we use a case study of 20 real-life PHEV models and 5 synthetic PHEV models whose battery capacities are much larger than those of the real-life models. The miles per gallon of gasoline (MPG), miles per gallon of gasolin-equivalent (MPGe), and capacities of each model are listed in Table 3. To obtain \(\eta_{j}^{\text{cd}}\) and \(\eta_{j}^{\text{cs}}\), we use the following conversions [27]:
\[
\text{KWh/km} \approx \frac{20.9433}{\text{MPGe}}, \quad \text{MPG} \approx 0.4251 \times \text{km/L}.
\]
The remaining vehicle characteristics are provided in Table 4. We use the same value of electricity price sensitivity and battery depletion cost as in [10], using the exchange rate 1 DKK = 0.13 EUR. The value of battery depletion cost is based on [28].

<table>
<thead>
<tr>
<th>j</th>
<th>Vehicle</th>
<th>MPGe</th>
<th>MPG (mi/gal)</th>
<th>$l_{max}^j$ (kWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25i+1</td>
<td>2020 Toyota Prius Prime</td>
<td>133</td>
<td>54</td>
<td>8.8</td>
</tr>
<tr>
<td>25i+2</td>
<td>2020 Hyundai Ioniq Plug-in Hybrid</td>
<td>119</td>
<td>52</td>
<td>8.9</td>
</tr>
<tr>
<td>25i+3</td>
<td>2020 Honda Clarity Plug-in Hybrid</td>
<td>110</td>
<td>42</td>
<td>17</td>
</tr>
<tr>
<td>25i+4</td>
<td>2020 Kia Niro Plug-in Hybrid</td>
<td>105</td>
<td>46</td>
<td>8.9</td>
</tr>
<tr>
<td>25i+5</td>
<td>2020 Ford Fusion Energi Plug-in Hybrid</td>
<td>103</td>
<td>42</td>
<td>9</td>
</tr>
<tr>
<td>25i+6</td>
<td>2020 Kia Optima Plug-in Hybrid</td>
<td>101</td>
<td>41</td>
<td>9.8</td>
</tr>
<tr>
<td>25i+7</td>
<td>2020 Subaru Crosstrek Hybrid AWD</td>
<td>90</td>
<td>35</td>
<td>8.8</td>
</tr>
<tr>
<td>25i+8</td>
<td>2020 Chrysler Pacifica Hybrid</td>
<td>82</td>
<td>30</td>
<td>16</td>
</tr>
<tr>
<td>25i+9</td>
<td>2020 Mitsubishi Outlander PHEV</td>
<td>74</td>
<td>25</td>
<td>13.8</td>
</tr>
<tr>
<td>25i+10</td>
<td>2020 MINI Cooper SE Countryman All4</td>
<td>73</td>
<td>29</td>
<td>10</td>
</tr>
<tr>
<td>25i+11</td>
<td>2020 Karma Revero GT (21-inch wheels)</td>
<td>70</td>
<td>26</td>
<td>28</td>
</tr>
<tr>
<td>25i+12</td>
<td>2020 BMW 530e</td>
<td>69</td>
<td>27</td>
<td>12</td>
</tr>
<tr>
<td>25i+13</td>
<td>2020 Mercedes-Benz GLC350e 4matic</td>
<td>68</td>
<td>25</td>
<td>13.5</td>
</tr>
<tr>
<td>25i+14</td>
<td>2020 Audi Q5</td>
<td>65</td>
<td>27</td>
<td>14.1</td>
</tr>
<tr>
<td>25i+15</td>
<td>2020 BMW 530e xDrive</td>
<td>65</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>25i+16</td>
<td>2020 Polestar 1</td>
<td>58</td>
<td>26</td>
<td>34</td>
</tr>
<tr>
<td>25i+17</td>
<td>2020 Volvo XC90 AWD PHEV</td>
<td>55</td>
<td>27</td>
<td>11.6</td>
</tr>
<tr>
<td>25i+18</td>
<td>2020 Porsche Panamera 4 e-Hybrid</td>
<td>51</td>
<td>23</td>
<td>17.9</td>
</tr>
<tr>
<td>25i+19</td>
<td>2020 Bentley Bentayga</td>
<td>45</td>
<td>19</td>
<td>17.3</td>
</tr>
<tr>
<td>25i+20</td>
<td>2020 Porsche Cayenne Turbo S e-Hybrid</td>
<td>39</td>
<td>18</td>
<td>17.9</td>
</tr>
<tr>
<td>25i+21</td>
<td>synthetic PHEV 01</td>
<td>58</td>
<td>22</td>
<td>75</td>
</tr>
<tr>
<td>25i+22</td>
<td>synthetic PHEV 02</td>
<td>68</td>
<td>27</td>
<td>70</td>
</tr>
<tr>
<td>25i+23</td>
<td>synthetic PHEV 03</td>
<td>78</td>
<td>32</td>
<td>65</td>
</tr>
<tr>
<td>25i+24</td>
<td>synthetic PHEV 04</td>
<td>88</td>
<td>38</td>
<td>60</td>
</tr>
<tr>
<td>25i+25</td>
<td>synthetic PHEV 05</td>
<td>98</td>
<td>43</td>
<td>55</td>
</tr>
</tbody>
</table>

Table 3  Miles per gallon of gasoline-equivalent (MPGe), miles per gallon of gasoline (MPG), and the battery capacity of 25 PHEV models. MPGe is used to measure energy consumption of alternative fuel vehicles, such as PHEVs. The indices $i \in \{0, 1, 2, \ldots \}$ and $j \in \{1, 2, \ldots \}$ determine the number and type of vehicle, respectively. For instance, the vehicle indexed by $i = 2$ and $j = 51$ refers to a 2020 Toyota Prius Prime model and the vehicle indexed by $i = 3$ and $j = 77$ refers to a 2020 Hyundai Ioniq Plug-in Hybrid.

Source: [29].

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum battery SOC</td>
<td>$l_{min}^j$, $j \in \mathcal{J}$</td>
<td>0.30% × $l_{max}^j$</td>
<td>kWh</td>
</tr>
<tr>
<td>Maximum rate of charging station</td>
<td>$v_{max}^j$</td>
<td>11</td>
<td>kWh</td>
</tr>
<tr>
<td>Battery depletion cost</td>
<td>$\gamma_{cd}^j$, $j \in \mathcal{J}$</td>
<td>5.299 × 10$^{-2}$</td>
<td>EUR/kWh</td>
</tr>
<tr>
<td>Fuel cost</td>
<td>$\gamma_f$</td>
<td>1.569</td>
<td>EUR/L</td>
</tr>
<tr>
<td>Electricity price sensitivity</td>
<td>$\alpha$</td>
<td>2.4 × 10$^{-8}$</td>
<td>(EUR/kWh)/kWh</td>
</tr>
<tr>
<td>Charging efficiency</td>
<td>$\eta_{ch}^j$, $j \in \mathcal{J}$</td>
<td>90%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4  Parameters of the 20 real-life PHEV models.
6.1. Constructing driving patterns

For testing, we construct a number of artificial yet realistic driving patterns of the fleet, i.e. stochastic processes of driving demands over time. For simplicity, we assume that the driving patterns of the vehicles are independent and identically distributed.

For each vehicle and each weekday, we consider a trip to work. The departure time of such trip follows a skewed Normal distribution, with location, scale, and skewness parameter set to 410 (this corresponds to 06:50 in the 24-hour notation), 20, and 8, respectively. The distribution is then truncated to the time interval between 03:00 and 12:00. The driving distance of the trip is Normally distributed $N(40.1, 2^2)$ and truncated to the interval between 30km and 60km. The probability density functions for departure time and driving distance are displayed in the left part of Figure 2. The average driving speed for the trip follows a Normal distribution $N(\mu, \sigma^2)$, truncated to the interval between 15km/h and 100km/h. Based on [30], we estimate the mean $\mu$ and standard deviation $\sigma$ from driving distances, using the functions in the right part of Figure 2. Once the departure time and the average driving speed are determined, we obtain the arrival time. Upon arrival, with a probability $\xi \in [0, 1]$, each vehicle makes another trip. We use the estimate $\xi = 0.3$. Departure time, distance, and speed have the same distributions with different parameters. We simply assume that vehicles have the same driving pattern every day from Monday to Friday.

On weekends, a driving pattern is likewise composed of either one or two trips, and we construct departure time, driving distance, and the average driving speed of each trip in the same manner. Similarly, we assume the driving patterns on Saturdays and Sundays are the same. Figure 3 illustrates the distribution of driving patterns over time. Note, however, that these stochastic processes of driving patterns may not be Markovian. In spite of this, we use this assumption as an approximation in the application of ADP.

6.2. Predicting electricity spot prices

We consider trading in the electricity spot market and specify the stochastic process of spot prices \( \{p_t : t = 0, 1, \ldots, T - 1\} \) as follows. We let the price consist of a deterministic and a random component such that $p_t = f_p(t) + g_t$, where $f_p(\cdot)$ is a known function accounting for seasonality and, for simplicity, \( \{g_t : t = 0, 1, \ldots, T - 1\} \) follows a first order autoregressive process, AR(1):

$$g_t = \phi_0 + \phi_1 \cdot g_{t-1} + \varepsilon_t^p, \ t = 1, 2, \ldots,$$

where $\varepsilon_t^p, t = 1, 2, \ldots$ are independent and identically Normally distributed $N(0, \sigma_p^2)$, and $\sigma_p \in (0, \infty)$, $\phi_0 \in \mathbb{R}$ and $\phi_1 \in (-1, 1)$ are parameters. Thus, $g_{t+1} | g_0, g_1, \ldots, g_t$ has the same distribution as $g_{t+1} | g_t$ (namely the distribution of $\varepsilon_{t+1}^p$), and so, prices are Markovian. In other words, the next
price depends only on the current price and not on the entire history of prices. It is straightforward to apply conditional sampling, i.e., a sample of \( g_t \) is generated from \( g_{t-1} \) and a random sample of \( \varepsilon_t^p \).

We consider the Nordic electricity spot market, Elspot, at Nord Pool and estimate the parameters using system prices from the selected period of 2021-01-11 to 2021-02-21. Historical prices, weekly seasonality, and deseasonalized prices are depicted in Figure 4. Weekly seasonality refers to the average hourly prices throughout a week. The seasonal value of the price for the hour 00:00 to 01:00 on Monday, for instance, is the average over all hours 00:00 to 01:00 on Mondays during the selected 6 weeks. Using ordinary least squares, the resulting parameter estimates are \( \hat{\phi}_0 \approx 5.40 \).
Figure 4  Historical spot prices of electricity from 2021-01-11 to 2021-02-21 (top), seasonality of electricity spot prices (bottom left) and historical deseasonalized spot prices of electricity from the selected period (bottom right).

with asymptotic standard error $\approx 0.727$, $\hat{\phi}_1 \approx 0.890$ with asymptotic standard error $\approx 0.0143$, and $\hat{\sigma}_p^2 \approx 29.5$. As seen from the figures, prices usually peak in the morning and late afternoon and dip during nighttime; they are higher on weekdays than on weekends, but are highly volatile with occasional spikes.

7. Computational results

We let $T$ be the number of hours in a week, i.e., $T = 168$, and construct a training data set with $J = 200$ and $N = 5000$ which consists of sample paths of state information $(l_n^0, \ldots, l_n^{T-1}, d_n^0, \ldots, d_n^{T-1}, p_n^0, \ldots, p_n^{T-1}, \bar{p}^n) \in \mathbb{R}^{(2 \times 200 + 1)T + 1}, n = 1, \ldots, 5000$. The expected future electricity price at time $T$ is simply set to the electricity price at time $T - 1$, i.e., $\bar{p}^n = p_{T-1}^n$. We use this training data to determine a policy $\hat{\pi}$.

We likewise construct a test data set with $J = 200$ and $N = 300$ which is independent of the training data set and composed of sample paths of driving demand and electricity prices only, i.e., $(d_n^0, \ldots, d_n^{T-1}, p_n^0, \ldots, p_n^{T-1}, \bar{p}^n) \in \mathbb{R}^{(200 + 1)T + 1}, n = 1, \ldots, 300$, with $\bar{p}^n = p_{T-1}^n$.

Algorithm 1 is run on an Apple M1 chip with 16G RAM and solves the minimization problem in Step 1.1 using the Gurobi Optimizer version 9.1.2.
7.1. Convergence

We investigate convergence of the LSMC algorithm for $J = 100$. For a varying number of training samples, we determine a policy and record the training time. The policy $\hat{\pi}$ is evaluated using all sample paths of the test data set, i.e., $N = 300$, by recording average total cost

$$\frac{1}{300} \sum_{n=1}^{300} \left( \sum_{t=0}^{T-1} C_t(S_t^n, A_t^n(S_t^n)) + C_T(S_T^n) \right).$$

Figure 5 illustrates the training time and so-called the learning curve of the LSMC algorithm.

As expected, the running time for training increases linearly with the number of samples. For $N = 5000$, the running time is around 40 minutes. Moreover, the LSMC-based policies demonstrate convergence as the number of samples increases. From $N = 1000$ to $N = 2000$, average total costs decrease by approx. 5%, whereas from $N = 4000$ to $N = 5000$, costs are almost constant.

The polices trained with varying number of samples are also evaluated on a single test path in terms of cumulative costs. The first panel of Figure 6 displays cumulative total costs for $j = 1, \ldots, 100$, i.e., $\sum_{s=0}^{t} \sum_{j=1}^{100} C_j(S_s^n, A_s^n(S_s^n))$. The other four panels show the costs $\sum_{s=0}^{t} C_j(S_s^n, A_s^n(S_s^n))$ for $j = 1, 10, 20, 25$, respectively. Generally, the policy trained using the largest data set produces the lowest accumulated cost over time. Also, the larger the data set, the less decrease in costs per increase in the number of samples, confirming the convergence.

We note that our LSMC method follows the algorithm of [21] in which the error between the value approximations and the true value functions is known to grow exponentially with $t$ due to accumulation, see [21, 31]. Here, we need sufficient training data to prevent the value function approximation for small $t$ from exploding as $t$ becomes larger.
7.2. Policy performance

We continue to analyze the performance of the policy for a fleet of $J = 100$ vehicles. Figures 7 - 10 illustrate charging and discharging behavior of the LSMC-based policy, obtained from the training data set with $N = 5000$ and for $j = 1, 10, 20, 25$. We evaluate the policy on a single test sample (the same as in Section 7.1) of exogenous electricity spot prices ($p_t$) and driving demands ($d_t$), as seen in the first and the third panels of the figures, respectively. The second panel displays the endogenous electricity prices reflecting the total load of the vehicle fleet. The fourth panel shows battery discharging $a_{j,t} \leq 0$ at times of driving $d_{j,t} > 0$ and not using the engine, and occasional charging $a_{j,t} > 0$ at times without demand $d_{j,t} = 0$. The fifth and sixth panels provide the corresponding battery level and the use of the engine, respectively. Finally, the seventh panel displays cumulative...
costs. Note that the real-life vehicle models $j = 1, 10, 20$ have decreasing CD and CS efficiencies and limited battery capacities, whereas the capacity of the synthetic model $j = 25$ is 3-5 times higher.

The charging policies of $j = 10, 20, 25$ are seen to fulfill the primary purpose of the battery: To supply electric power to the vehicle. In fact, the battery is always charged (see positive values in the fourth panels of Figures 8 - 10) such as to fill the battery (cf. battery levels in the fifth panels) prior to a trip (see driving demand in the third panels) and thereby facilitate driving in CD mode (cf. negative values in the fourth panels) as far as possible. For $j = 10$ and $t = 138$, for example, the vehicle discharges the battery (fourth panel, Figure 8), which was charged to its maximum level at $t = 135$ (fourth and fifth panels, Figure 8). For $t = 153$, however, the battery capacity (fifth panel, Figure 8) is insufficient to fully cover driving demand (third panel, Figure 8). Given that discharging is less costly than using the engine, the vehicle should indeed favor CD mode over CS mode, unless the following battery level will be at its minimum. This is also clearly confirmed by $j = 25$, which never has to use the engine (sixth panel, Figure 10) because of its battery size. For $j = 1$ and $t = 138$, however, the vehicle drives in CS mode (see use of the engine in the sixth panel, Figure 7) even though the following battery level will be above its minimum (fifth panel, Figure 7). Thus, the vehicle has higher fuel consumption than necessary, which has a significant impact on total costs. The reason for such non-optimal behavior may be that the ADP requires more training data or more flexibility in the approximations.

In addition to driving demands, the charging policy must adapt to the development in electricity prices (cf. first and second panels of Figures 7 - 10). Nevertheless, charging does not always occur at the times of low prices. For $j = 10$ and $t = 25$, for example, the vehicle charges (fourth panel, Figure 8) such as to meet driving demand at $t = 31$ (third panel, Figure 8), although the price is lower at $t = 27$ (first and second panels, Figure 8). There may be various potential reasons for this: If all vehicles charge at the same time, the electricity price increases, and thus, the exogenous price may be low but the actual endogenous prices may be higher. Some of the vehicles should, however, charge when the exogenous price is low. A comparison of exogenous and endogenous prices, however, fails to support this explanation. Another reason is that the value function approximation is inaccurate. As a result of an inaccurate second-stage value function, the first-stage action deviates from the true optimal solution, making the second-stage state inaccurate, and thus, the second-stage action deviates even more. Thus, approximation errors may accumulate over time. Finally, a highly likely explanation is related to uncertainty. Note that we evaluate the policy on a single price path rather than the entire distribution of future prices. At each stage, the optimal action should be determined by the trade-off between the current and expected future value. The single price path, however, may not predict the expected future price.
7.3. Immediate charging

We compare the LSMC-based policy to a simple rule-based policy for charging and discharging. The simple rule-based policy determines an action as follows:

\[
    a_{j,t} = \begin{cases} 
    \min\{l_{j,t}^{\max} - l_{j,t}, v_{j,t}^{\max}\}, & \text{if } d_{j,t} = 0, \\
    -\min\{l_{j,t}^{\min} - l_{j,t}, \eta_{j,t} \cdot d_{j,t}\}, & \text{if } d_{j,t} > 0.
    \end{cases}
\]

In other words, the simple policy charges the battery immediately after the trip is finished and discharges the battery until it reaches its minimum SOC. This fulfills the primary purpose of the battery in the sense that the policy promotes driving in CD mode as much as possible. The simple policy, however, may charge the battery when the electricity price is high, as it does not consider the development in prices.

In this comparative study, we train the LSMC-based policy using the training data set with \( N = 5000 \) and evaluate both policies on the entire test data set with \( N = 300 \). In addition to the
Lee and Boomsma: An approximate dynamic programming algorithm for short-term electric vehicle fleet operation under uncertainty

Article submitted to Applied Energy

<table>
<thead>
<tr>
<th>j</th>
<th>100, N = 5000, j = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day-ahead spot prices of the electricity (p_t)</td>
<td></td>
</tr>
<tr>
<td>Electricity price including load from vehicles (Δ_l)</td>
<td></td>
</tr>
<tr>
<td>Driving demands (d_{jt})</td>
<td></td>
</tr>
<tr>
<td>Actions (a_j)</td>
<td></td>
</tr>
<tr>
<td>Battery levels (q_j)</td>
<td></td>
</tr>
<tr>
<td>Gasoline usage</td>
<td></td>
</tr>
<tr>
<td>Cumulative Costs (\bar{C}_t)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8**  Performance of the LSMC-based policy for the vehicle j = 10 out of a fleet of J = 100.

<table>
<thead>
<tr>
<th>j</th>
<th>100, N = 5000, j = 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day-ahead spot prices of the electricity (p_t)</td>
<td></td>
</tr>
<tr>
<td>Electricity price including load from vehicles (Δ_l)</td>
<td></td>
</tr>
<tr>
<td>Driving demands (d_{jt})</td>
<td></td>
</tr>
<tr>
<td>Actions (a_j)</td>
<td></td>
</tr>
<tr>
<td>Battery levels (q_j)</td>
<td></td>
</tr>
<tr>
<td>Gasoline usage</td>
<td></td>
</tr>
<tr>
<td>Cumulative Costs (\bar{C}_t)</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 9**  Performance of the LSMC-based policy for the vehicle j = 20 out of a fleet of J = 100.
average total costs, Figure 11 provides the average price for purchasing electricity and the average consumption of gasoline, computed as

\[
\text{average electricity purchase price} = \frac{\sum_{n=1}^{300} \sum_{t=0}^{T-1} \sum_{j} d_{j,t}^{n} \hat{p}^n_{j,t} \cdot a_{j,t}^{n,\pi} \cdot \frac{1}{\eta_j^{t}}}{\sum_{n=1}^{300} \sum_{t=0}^{T-1} \sum_{j} d_{j,t}^{n} a_{j,t}^{n,\pi} \cdot \frac{1}{\eta_j^{t}}},
\]

\[
\text{average gas consumption} = \frac{1}{300} \sum_{n=1}^{300} \sum_{t=0}^{T-1} \sum_{j} d_{j,t}^{n} \left( d_{j,t}^{n} + a_{j,t}^{n,\pi} \cdot \frac{1}{\eta_j^{t}} \right),
\]

where \(a_{j,t}^{n,\pi}\) is the action determined by the trained policy \(\pi\) for vehicle \(j\) at time \(t\) and on the \(n\)-th test sample, and \(\hat{p}^n_{j,t}\) is the endogenous electricity price, including the load of the fleet at time \(t\) in the \(n\)-th test sample. The average purchase price is the total electricity costs over all 300 test samples divided by the total electricity consumption over the entire test data set. The average gas consumption is the total use of gas over all 300 test samples set divided by the size of test data set, i.e. 300.

The upper panel of Figure 11 illustrates that the LSMC-based policy outperforms the simple policy for small to medium-sized fleets, i.e. \(J = 5, 10, 25\). The increase in costs by using the simple policy, however, is less than 2%. For larger fleet sizes, \(J = 50, 75, 100\), the simple rule-based policy performs better, reducing costs by up to 4%.

Figure 10 Performance of the LSMC-based policy for the vehicle \(j = 25\) out of a fleet of \(J = 100\).
These results are consistent with our observations in Section 7.2. On the one hand, the LSMC-based policy seeks to charge at low prices and outperforms the simple policy in this respect, as seen in the lower left panel of Figure 11. On the other hand, the LSMC-based policy does not always drive in CD mode as far as possible and often fails to charge the battery prior to a trip. This leads to higher average gasoline consumption than for the simple policy, as confirmed by the lower right panel of Figure 11. For larger fleet sizes, the costs of gasoline outweigh the value of favorable prices. In particular, the ability of the LSMC-based policy to charge at low prices and favor CD mode over CS mode decreases with the number of vehicles.

Such behavior of the LSMC-based policy may be due to insufficient training of the model or lack of model flexibility when the fleet size is large, i.e. when the state and action spaces have high dimensionality. A training set with $N = 5000$ samples may not be sufficient to cover the majority of the state space, and hence, the LSMC-based policy may not fully explore this. For better performance, one could generate simulation paths more efficiently, see e.g. [25], or increase model flexibility by using non-linear approximations. Note also that without the sensitivity of the price with respect to load, the fleet problem decomposes according to vehicles. As a result, the
optimal charging problem can be solved independently for each vehicle using exogenous prices, and hence, with a state space of manageable dimensionality.

Alternatively, the failure of the LSMC-based policy to charge the battery prior to a trip may be due to the uncertainty of future driving demands, including random departure times. To further investigate this, we carry out an experiment with deterministic driving patterns, see Figure 12. In this experiment, the average consumption of gasoline under the LSMC-based policy is reduced, which makes this policy outperform the simple policy. The increase in costs by using the simple policy is up to 4%.

We return to uncertainty in driving patterns and continue to inspect conditions under which the LSMC-based policy improves performance. With a very small sensitivity of the price with respect to load, each vehicle is almost independent of the remainder of the fleet, requiring a substantial number of simulation paths to explore the state and action spaces. With stronger co-dependencies of the vehicles, however, the LSMC-based policy may require fewer samples. To examine this, we increase $\alpha$. An increase of a factor 1000 does not make the LSMC-based policy outperform the simple policy, see the appendix. The results of increasing $\alpha$ by a factor 2000 are shown in Figure 13.
The upper panel illustrates that the LSMC-based policy now outperforms the simple policy for all fleet sizes. Although the average gasoline consumption remains higher under the LSMC-based policy, this is offset by a lower average purchasing price for electricity. The increase in costs by using the simple policy is approximately 2%.

7.4. Scalability

The dynamic programming method may suffer from scalability problems due to the-curse-of-dimensionality with regard to state and action spaces. With $J$ vehicles in the fleet, the dimension of the state space is $2J + 1$. Discretizing the battery level of each vehicle into $I$ intervals produces $I^J$ battery states in total. Similarly, discretizing the driving demand of each vehicle into $I$ intervals produces $I^J$ driving states in total. For each of the $I^{2J}$ combinations of battery level and driving demand, we may have up to $I^{4J}$ discrete actions available.

The ADP approach seeks to handle this curse-of-dimensionality. First, with simulation, we may work with $N << I^{2J}$ sample states. Second, with a linear approximation, we may solve $N$ linear programming problems as opposed to complete enumeration of $I^{4J}$ actions. Figure 14 investigates...
the scalability of the LSMC method with respect to the size of vehicle fleet, $J$. Given a fixed number of training samples, the time for training obviously increases with the number of vehicles. The speed of increase, however, is rather low, allowing us to solve the charging problem with a large number of vehicles.

8. Conclusion

This paper develops a dynamic and stochastic optimization problem for short-term electric vehicle operation and adopts an approximate dynamic programming approach referred to as the least square Monte Carlo method. Considering the sizes of state and action spaces, simulation and value function approximation allows for scaling with respect to the number of different vehicles. Although the performance of the LSMC-based policy should be improved for large-scale applications, scalability assessments demonstrate the potential of the ADP approach. In fact, we are able to establish convergence of our algorithm for 100 vehicles by using 5000 samples in the simulation, with a runtime of around 40 minutes. The LSMC-based policy requires the vehicles to discharge the battery rather than using the engine, as far as possible. The timing of battery charging, however, should be adapted to the development in power prices. Indeed, for small and medium-sized fleets, we find the costs of a simple immediate charging policy to be approximately 2% and 4% higher than the LSMC-based policy in stochastic and deterministic settings, respectively.

With a linear value function approximation, our method is straightforward to extend to more detailed models for electric vehicle operation. For example, vehicles may decide not to meet their

Figure 14  Training of LSMC as a function of the number of vehicles.
driving needs (maximizing the utility of demand), or the aggregator may decide not to satisfy demand (minimizing the costs of load curtailment). A more involved extension may relax the assumption that the aggregator has full control of the vehicles. If the aggregator rather anticipates the optimal actions of the vehicle owners, the model should ideally be bi-level.

From a methodological perspective, it may be relevant to compare the post-decision problem to the pre-decision formulation, not least since the linear value function approximation allows for explicit computation of the expected future pre-decision value. Alternatively, a so-called regress-later strategy may outperform the regress-now. Finally, it may be relevant to investigate other value function approximations, for instance by iteratively approximating the value function or the policy.

Acknowledgements

S. Lee and T. K. Boomsma gratefully acknowledge support from the project Analyses of Hourly Electricity Demand (AHEAD) funded by ForskEl 2017.

9. Appendix
Figure 15 Average total costs of the LSMC-based policy and the simple rule-based policy (top), average purchase price of the electricity (bottom left) and average gasoline consumption (bottom right), with $\alpha$ set to $2.4 \times 10^{-5}$.
References


