Reinforcement Learning of Heuristic EV Fleet Charging in a Day-Ahead Electricity Market

Stijn Vandael, *Member, IEEE*, Bert Claessens, Damien Ernst, *Member, IEEE*, Tom Holvoet, *Member, IEEE*, and Geert Deconinck, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of defining a day-ahead consumption plan for charging a fleet of electric vehicles (EVs), and following this plan during operation. A challenge herein is the beforehand unknown charging flexibility of EVs, which depends on numerous details about each EV (e.g., plug-in times, power limitations, battery size, power curve, etc.). To cope with this challenge, EV charging is controlled during opertion by a heuristic scheme, and the resulting charging behavior of the EV fleet is learned by using batch mode reinforcement learning. Based on this learned behavior, a cost-effective day-ahead consumption plan can be defined. In simulation experiments, our approach is benchmarked against a multistage stochastic programming solution, which uses an exact model of each EVs charging flexibility. Results show that our approach is able to find a day-ahead consumption plan with comparable quality to the benchmark solution, without requiring an exact day-ahead model of each EVs charging flexibility.**

***Index Terms*—Demand-side management, electric vehicles (EVs), reinforcement learning (RL), stochastic programming (SP).**

Nomenclature

The symbols and notations used throughout this paper are summarized below.

Sets

$^{i}\theta_D$ Set of all charging parameters available at day D for electric vehicle (EV) i.

$^{i}\Omega_t$ Set of charging parameters available at time t for EV i.

$\mathcal{S}$ Set of scenarios.

s_n Set of charging parameters in scenario n.

Parameters

H Total number of market periods in a day.

T Total number of control periods in a day.

Δt Length of a control period.

N_{s} Number of scenarios.

N_{ev} Number of EVs in an EV fleet.

N_{ctrl} Number of control periods in a market period.

λ_h Price in the day-ahead market for market period h.

λ_h^- Negative imbalance price for market period h.

λ_h^+ Positive imbalance price for market period h.

$^{i}T_{\mathrm{arr}}$ Arrival time of EV i.

$^{i}T_{\mathrm{dep}}$ Departure time of EV i.

$^{i}E_{\mathrm{req}}$ Requested energy of EV i.

$^{i}P_{\mathrm{lim}}$ Charging power limit of EV i.

P_{grid} Maximum total charging power of the EV fleet.

π^n Probability of scenario n.

$^{i}t^n_{\mathrm{start}}$ First control period of EV i in scenario n.

$^{i}t^n_{\mathrm{end}}$ Final control period of EV i in scenario n.

$^{i}E^n_{\mathrm{req}}$ Required energy of EV i in scenario n.

$\Delta\tau$ Temperature step in Boltzmann exploration.

$\mathbf{f}_s$ Simultaneity factor.

β Offset from day-ahead prices, to define imbalance prices.

Functions

$\mathbf{T}(h)$ Mapping from market period h to the set of control periods in market period h.

f_{heur} Heuristic function to dispatch power to EVs.

Real variables

E_h^{da} Energy bought in the day-ahead market for market period h.

Manuscript received October 15, 2013; revised March 20, 2014 and August 28, 2014; accepted October 20, 2014. Date of publication March 9, 2015; date of current version June 18, 2015. This work was supported in part by the DistriNet Research Group of the Department of Computer Science, Catholic University of Leuven, in part by Vlaamse Instelling Voor Technologisch Onderzoek, Flemish Institute for Technological Research, in part by the Electrical Energy and Computing Architectures Research Group of the Department of Electrical Engineering, Catholic University of Leuven, and in part by the Department of Electrical Engineering and Computer Science, University of Liège. B. Claessens conducted research for this paper in the context of the Linear project. The work of S. Vandael was supported by the Institute for the Promotion of Innovation by Science and Technology in Flanders. Paper no. TSG-00787-2013.

S. Vandael is with the Department of Computer Science, Katholieke Universiteit Leuven, 3001 Leuven, Belgium, and also with Energyville, 3600 Genk, Belgium (e-mail: stijn.vandael@cs.kuleuven.be).

B. Claessens is with the Department of Energy, Research Institute Vlaamse Instelling Voor Technologisch Onderzoek, 2400 Mol, Belgium, and also with Energyville, 3600 Genk, Belgium.

D. Ernst is with the Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium.

T. Holvoet is with the Department of Computer Science, Katholieke Universiteit Leuven, 3001 Leuven, Belgium.

G. Deconinck is with the Department of Electrical Engineering, Katholieke Universiteit Leuven, 3001 Leuven, Belgium, and also with Energyville, 3600 Genk, Belgium.

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

Digital Object Identifier 10.1109/TSG.2015.2393059

E'_h	Energy charged by the EV fleet in market period h.
${}^i x_t$	Energy charged by EV i in control period t.
${}^i a_t$	Charging power of EV i during control period t.
${}^i P_t^{\text{ctrl}}$	Power requested from EV i for control period t.
P_t^{da}	Power requested from the fleet for control period t.
${}^i P_t^{\text{min}}$	Minimum charging power of EV i for time t.
${}^i P_t^{\text{max}}$	Maximum charging power of EV i for time t.
${}^i \tau$	Heuristic priority value of EV i.
z_t	Energy charged by the EV fleet in control period t.
${}^i P_t^n$	Charging power of EV i during control period t in senario n.

I. Introduction

NOWADAYS, controlled EV charging is a popular research topic [1]. This trend is driven by two factors: 1) the significant charging flexibility of EVs, which are idle during a large part of the day and 2) the decreasing controllability of electricity generation due to the rapid increase of renewables. In a liberalized electricity market, aggregators are typically seen as the actors who will utilize the flexibility of EVs [2]. For an aggregator, algorithms and models for controlled charging of EVs are important to efficiently optimize its provision of ancillary services [3], [4], or its energy trading activities [5]. In this paper, we focus on the latter case, where an aggregator purchases EV charging energy in the day-ahead market, and incurs imbalance costs in the imbalance market.

To define a day-ahead consumption plan for EVs and follow this plan during operation, an aggregator requires information about the charging flexibility of its EV fleet. However, this flexibility is subject to human behavior, and not necessarily all technical information about a privately owned EV is readily available. In current charging standards [6], [7], only a limited set of parameters is communicated between EV and aggregator (e.g., current battery level, maximum charging power). Therefore, it can be difficult to construct an accurate mathematical model of an EVs charging flexibility. Driven by this challenge, we propose a "blind" learning approach which does not require any prior knowledge. In this approach, individual EV charging is controlled by a heuristic scheme which only uses readily available parameters, while a reinforcement learning (RL) approach learns the resulting collective charging behavior of the EV fleet. Based on this learned charging behavior, a cost-effective day-ahead plan can be defined. Summarized, the contributions of this paper are as follows.

1) Description of a RL approach to learn EV charging behavior, which is determined by a predefined heuristic scheme.
2) Evaluation of the RL approach through benchmarking against a multistage stochastic programming (SP) method, which uses an exact model. This evaluation shows that our approach is able to reach a near-optimal solution in absence of an exact model of the EV fleet.

In Section II, an overview of related work is presented. In Section III, the considered problem of an EV aggregator is described in detail. In Section IV, our RL approach to this problem is described. In Section V, the RL approach is benchmarked against a SP solution, and evaluated in a large-scale realistic scenario of an EV fleet in Belgium.

II. Related Work

Related work of this paper is divided in two parts. In the first part, we give an overview of papers which describe algorithms to improve day-ahead planning. In the second part, an overview is given of papers which describe RL algorithms for demand response (DR).

A. Day-Ahead Planning

In most work concerning day-ahead planning of generation and loads, an exact mathematical model is assumed available. In our approach, which does not assume beforehand knowledge of a model, these planning methods are used as a benchmark.

Al-Awami and Sortomme [8] formulated the problem of day-ahead balancing of vehicle-to-grid (V2G) services with wind and thermal energy as a mixed-integer stochastic linear program. The stochastic variables in this problem are the wind power generation, market prices, and imbalance prices. Simulation results show that coordination based on this model can increase expected profits while improving the conditional value at risk. In our problem description, we assume that a model of the EV fleet is not readily available.

Plazas *et al.* [9] and Caramanis and Foster [10] proposed SP methodologies for optimal bidding in multiple markets. Examples of stochastic variables identified in the described problems are clearing prices, number of available plug-in hybrid EVs, required charging energy, etc. While these papers capture the complex interactions between multiple markets, we focus on predicting the load for a day-ahead market, without assuming prior knowledge about available EVs. SP is used as a benchmark for our approach (Section V).

Wu *et al.* [5] proposed an algorithm for day-ahead load scheduling, and a dynamic dispatch algorithm for distributing purchased energy to plug-in electric vehicles (PEVs). In this algorithm, electricity prices and PEV charging behavior are considered deterministic. Simulation results show that the dispatched load perfectly matches the purchased energy. In our problem description, EV charging behavior is assumed to be nondeterministic and unknown beforehand.

B. Reinforcement Learning for DR

In this paper, we use RL to learn the heuristic behavior of an EV fleet. An important challenge in RL is dealing with continuous and very large state and action spaces [11]. In this section, a representative selection of RL papers for DR is given, and we briefly explain how these papers deal with large spaces.

Lee and Powell [12] proposed a bias-corrected form of Q-learning to operate battery charging. This correction is introduced to cope with the bias toward overestimated Q-values, induced by the max-operator. This bias is a well-know problem in Q-learning, and the authors report that this issue exacerbates in the presence of highly volatile prices, which cause large overestimates. In evaluation, a scenario of a 10 MWh battery, and a price-model based on real world spot prices is used. Simulations of this scenario show that bias-corrected Q-learning significantly reduces the bias, and learns a better policy compared to classic Q-learning. Both state space and action space are relatively small for one battery. Nonetheless, a significant amount of iterations are necessary ($\sim 10^6$).

Shi and Wong [13] proposed a RL approach to provide V2G services by a fleet of EVs. The used Markov decision process (MDP) is considered from the viewpoint of one EV, i.e., no coupling constraints (constraints which involve actions of multiple EVs) are included. The source of uncertainty in this MDP is the electricity price, which is modeled as a two-state Markov chain with unknown transition probabilities. Simulation results show that profit can significantly be increased for EV owners. The size of the state and action space is kept small by using an MDP for each EV, which are independent from each other in absence of coupling constraints. In our problem description, control actions have to be coordinated among EVs to follow a collective day-ahead schedule.

Levorato *et al.* [14] proposed a RL approach to adjust energy consumption of an individual residential consumer. In this approach, both energy prices and consumer decisions are modeled as an MDP. The structure and transition probabilities of these MDPs are unknown, and need to be learned. In simulations, this approach was able to reduce a consumer's costs by 16%–40% compared to the uncontrolled case. Because only one consumer is considered, state and action spaces are limited in size.

Several papers propose RL techniques in electricity markets. In [15] and [16], RL approaches are proposed for learning bidding strategies in forward electricity markets. Reddy and Veloso [17] proposed a RL approach to learn pricing strategies for a broker agent in a tariff market. In this paper, the authors report a state space of more than 10^{12} states for five brokers at two tariff prices each, and use simple heuristics to reduce the state space.

In our approach, we drastically reduce the state and action space by defining an MDP over the whole EV fleet. Rather than using individual EV control actions (e.g., charge EV 2 at 3 kW), we use collective EV fleet control actions (e.g., charge the EV fleet at 2 mW). To translate collective control actions back to individual control actions, a simple heuristic is used. Based on historic data of collective control actions, a cost-effective day-ahead plan is learned, which inherently takes into account the heuristic division strategy. Furthermore, to deal with continuous variables in our state and action space, we use fitted Q iteration [18] instead of temporal difference learning [19]. This advanced technique allows us to deal with continuous spaces, and generalize over different observations.

III. Aggregator Problem Description

The main stakeholder in our problem description is an aggregator, who manages a fleet of EVs. The decisions made by an aggregator are divided in two decision phases (Fig. 1).

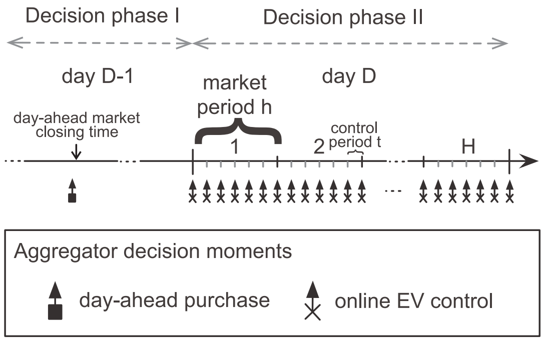

Fig. 1. Day-ahead purchase of EV charging energy.

In the first decision phase (day D-1), the aggregator predicts the energy required for charging its EVs for day D, and purchases this amount in the day-ahead market. During the second decision phase (day D), the aggregator communicates with the EVs to control their charging, based on the amount of energy purchased in the day-ahead market during the first decision phase.

A. Decision Phase I

In the first decision phase (day D-1), the aggregator decides how much energy he purchases in the day-ahead market. In the day-ahead market, energy is purchased for each market period of day D, the next day. The purchase of an aggregator can be summarized in a day-ahead load schedule

$$\mathbf{E}^{\mathrm{da}} \triangleq \left\{E_1^{\mathrm{da}}, \ldots, E_H^{\mathrm{da}}\right\} \tag{1}$$

with H the total number of market periods in a day. The length of a market period h and the market closing time are dependent on the considered day-ahead market. Once the day-ahead market closes at day D-1, no more purchases can be made for the next day. An example of a day-ahead market is Belpex (Belgium), with hourly market periods and a market closing time at 12:00 A.M. In this paper, we assume the amount of energy which can be bought in a single market period is limited, based on a grid constraint P_{grid}. Detailed transformer and feeder limitations are not taken into account.

The decision of defining a day-ahead load schedule is driven by two factors. First, the costs of purchasing the load schedule in the day-ahead market should be minimized based on day-ahead prices, which are defined per market period h

$$\boldsymbol{\lambda} \triangleq \{\lambda_1, \ldots, \lambda_H\}. \tag{2}$$

In this paper, we assume predictions of day-ahead prices are available, which is supported by well-advanced day-ahead price prediction methods [20]. Furthermore, we assume the aggregator is a price-taker. In case of limited size purchase orders, an aggregator will naturally have a price-taker position.

Second, imbalances in the load schedule are not allowed, i.e., the scheduled energy should be able to be charged by the EV fleet without imbalances in decision phase II (Section III-B). In this paper, we assume that an aggregator will never define a load schedule which intentionally causes imbalances. The motivations for this assumption are

TABLE I
TYPICAL EV STATE PARAMETERS

Parameter	description	unit
P_{min}	Minimum charging power	kW
P_{max}	Maximum charging power	kW
E_{req}	Requested energy	kWh
t_{dep}	Departure time	s

the restrictions on gaming and abuse of an electricity market. In terms of the latter motivation, Belpex market rules state [21]: "the participant guarantees the correctness and the accuracy of the orders that it submits on the trading platform." Furthermore, we assume that imbalance prices are unknown, because they are typically volatile and unpredictable.

At the end of the first decision phase, the load schedule $\mathbf{E}^{\mathrm{da}}$ in (1) has been purchased at the day-ahead market for day D.

B. Decision Phase II

During day D, the aggregator has the opportunity to communicate with the EVs in order to control their charging power. Typically, online controlling EVs will happen on a shorter time scale than market orders. In this paper, we divide each market period h in a number of equally spaced control periods. For each control period t, the power requests of an aggregator for each EV in its fleet can be summarized in

$$\mathbf{P}_t^{\mathrm{ctrl}} \triangleq \left\{ {}^{1}P_t^{\mathrm{ctrl}}, \ldots, {}^{N_{\mathrm{ev}}}P_t^{\mathrm{ctrl}} \right\} \quad \forall t \in \{1, \ldots, T\}. \tag{3}$$

Based on these requests, the grid-connected EVs locally decide upon their actual charging power ${}^{i}a_t \in \mathbf{a}_t$, where $\mathbf{a}_t = \pi(\mathbf{P}_t^{\mathrm{ctrl}})$. In this paper, we define π as a policy function which assures the user requirements on the battery state of charge are respected, while following the aggregator's requested control power ${}^{i}P_t^{\mathrm{ctrl}}$ as closely as possible. Based on all requested power values ${}^{i}P_t^{\mathrm{ctrl}}$, the energy charged by the EV fleet in each market period h is

$$E'_h = \sum_{t \in \mathbf{T}(h)} \sum \pi \left(\mathbf{P}_t^{\mathrm{ctrl}} \right) \Delta t \quad \forall h \in \{1, \ldots, H\} \tag{4}$$

where the function $\mathbf{T}(h) = \{(h-1)N_{\mathrm{ctrl}} + 1, \ldots, hN_{\mathrm{ctrl}}\}$ maps a market period onto its respective control periods.

For each control period, the aggregator decides the control power of each EV. To make this decision, we assume the aggregator can request the present state of all EVs right before each control period. This state is based on parameters found in current charging standards [6], [7] (Table I).

The decisions made for each control period are driven by the minimization of imbalances between the day-ahead load schedule $\mathbf{E}^{\mathrm{da}}$ in (1), and the actual load in (4). In case of a negative imbalance (more energy charged than bought at the day-ahead market, $E'_h > E_h^{\mathrm{da}}$), the aggregator has to pay extra, based on a negative imbalance price $\lambda_h^- > \lambda_h$. In case of a positive imbalance (less energy charged than bought in the day-ahead market, $E'_h < E_h^{\mathrm{da}}$), the aggregator gets refunded based on a positive imbalance price $\lambda_h^+ < \lambda_h$. Because positive imbalance prices are lower than day-ahead prices, the aggregator will only be partially refunded for its excess energy bought in the day-ahead market. Based on the imbalance prices, the complete cost function can be defined[1]

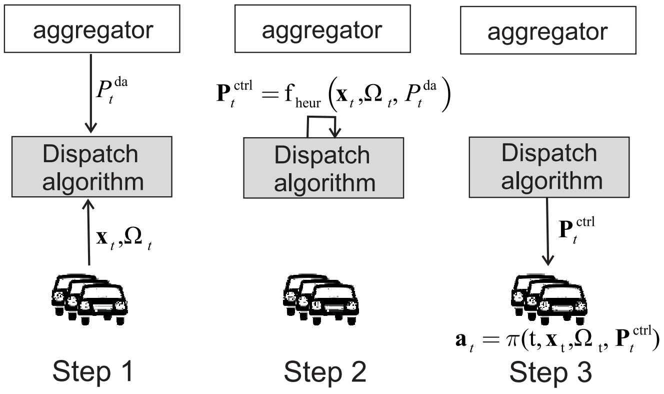
aggregator
aggregator
aggregator
P_t^{da}
$\mathbf{P}_t^{\mathrm{ctrl}} = \mathrm{f}_{\mathrm{heur}}\left(\mathbf{x}_t, \Omega_t, P_t^{\mathrm{da}}\right)$
Dispatch algorithm
Dispatch algorithm
Dispatch algorithm
$\mathbf{x}_t, \Omega_t$
$\mathbf{P}_t^{\mathrm{ctrl}}$
$\mathbf{a}_t = \pi(\mathrm{t}, \mathbf{x}_t, \Omega_t, \mathbf{P}_t^{\mathrm{ctrl}})$
Step 1
Step 2
Step 3

Fig. 2. EV charging control by the dispatch algorithm.

$$\sum_{h=1}^{H} \left\{ E_h^{\mathrm{da}} \lambda_h + \left[E'_h - E_h^{\mathrm{da}} \right]_+ \lambda_h^- - \left[E_h^{\mathrm{da}} - E'_h \right]_+ \lambda_h^+ \right\} \tag{5}$$

with

$$\lambda_h^- = \lambda_h + \beta \tag{6}$$

$$\lambda_h^+ = \lambda_h - \beta \tag{7}$$

where negative and positive imbalance prices are β higher and β lower, respectively, than the known day-ahead prices. The choice of β is based on the typical difference between day-ahead and imbalance price in the considered electricity market.

At the end of decision phase II, the aggregator knows the total imbalance costs to be paid for day D. In order to minimize these costs, together with day-ahead costs (decision phase I), the aggregator needs to learn the charging flexibility of its EV fleet.

IV. REINFORCEMENT LEARNING APPROACH

In this section, we present our RL approach to the aggregator problem formulated in Section III. A key challenge in this problem is the beforehand unknown charging flexibility of individual EVs. Rather than modeling individual EVs, our approach learns the collective heuristic charging behavior of the EV fleet.

A. Heuristic Online Control of the EVs (Decision Phase II)

In the second decision phase (day D), the aggregator controls the charging of its EVs to follow a day-ahead power schedule defined in decision phase I (Section IV-B):

$$\mathbf{P}^{\mathrm{da}} \triangleq \left\{ P_1^{\mathrm{da}}, \ldots, P_T^{\mathrm{da}} \right\}. \tag{8}$$

The aggregator follows this power schedule as closely as possible by using a dispatch algorithm in three steps (Fig. 2). In step 1, the dispatch algorithm gathers state information from the EV fleet, and a scheduled power value P_t^{da} from the aggregator. In step 2, the dispatch algorithm uses this information

[1] For a number $a \in \mathbb{R}$, $[a]_+$ denotes max[a, 0]

to calculate the control power values $\mathbf{P}_t^{\text{ctrl}}$ in (3). In step 3, each control power value is communicated to its respective EV, which takes the control power value as input to its local decision making process. Before explaining the dispatch algorithm in detail, this local decision making process is described.

The charging behavior of an EV i is based on its charging parameters $^i\theta_D$ during day D

$$\begin{aligned}&\forall i \in \{1, \ldots, N_{\text{ev}}\}\\ &{}^i\theta_D = \emptyset \vee \left({}^iT_{\text{arr}},\ {}^iT_{\text{dep}},\ {}^iE_{\text{req}},\ {}^iP_{\text{lim}}\right)\end{aligned} \tag{9}$$

where $^i\theta_D$ is empty when EV i does not charge during day D. Consequently, N_{ev} is a maximum bound on the EVs that can arrive during day D. Based on $^i\theta_D$, $^i\Omega_t$ contains the charging parameters available at time t for EV i

$$\begin{aligned}&\forall i \in \{1, \ldots, N_{\text{ev}}\}\\ &{}^i\Omega_t = \left\{{}^i\theta_D \mid t \geq {}^iT_{\text{arr}} \wedge t \leq {}^iT_{\text{dep}}\right\}.\end{aligned} \tag{10}$$

To model the local decision making of the EVs, their charging behavior is represented as an MDP. The state space X of the EVs is composed by the charged energy, and defined as

$$X = \left\{\mathbf{x} \in \mathbb{R}^{N_{\text{ev}}} \mid {}^ix \in \left[0,\ {}^iE_{\text{req}}\right]\right\} \tag{11}$$

where $^iE_{\text{req}}$ is its required amount of energy at departure time $^iT_{\text{dep}}$. A full charging cycle for an EV i starts in the initial state $x_t = 0$ at arrival time $^iT_{\text{arr}}$, and ends in the terminal state $x_t = {}^iE_{\text{req}}$ at departure time $^iT_{\text{dep}}$. The action space A is composed of all charging actions, defined as

$$A = \left\{\mathbf{a} \in \mathbb{R}^{N_{\text{ev}}} \mid {}^ia \in \left[0,\ {}^iP_{\text{lim}}\right]\right\} \tag{12}$$

with $^iP_{\text{lim}}$ the power limit defined by the EVs battery management system. The system dynamics of the EVs are described by the state transition

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{a}_t\Delta t \tag{13}$$

with Δt the length of a control period. These transitions are only possible between arrival and departure time of an EV. The policy π of the EVs is to charge their battery before departure time, while following the aggregator's requested charging power $\mathbf{P}_t^{\text{ctrl}}$ in (3) as closely as possible:

$$\mathbf{a}_t = \pi\left(t, \mathbf{x}_t, \Omega_t, \mathbf{P}_t^{\text{ctrl}}\right) \tag{14}$$

where $\Omega_t = \{^1\Omega_t, \ldots, {}^{N_{\text{ev}}}\Omega_t\}$. For each EV, this policy determines an action $^ia_t \in \mathbf{a}_t$, based on the charged energy ^ix_t, charging parameters $^i\Omega_t$, and requested charging power $^iP_t^{\text{ctrl}}$

$$\begin{aligned}&\forall i \in \{1, \ldots, N_{\text{ev}}\}\\ &{}^ia_t = \begin{cases} {}^iP_t^{\min}, & \text{if } {}^iP_t^{\text{ctrl}} < {}^iP_t^{\min}\\ {}^iP_t^{\text{ctrl}}, & \text{if } {}^iP_t^{\min} \leq {}^iP_t^{\text{ctrl}} \leq {}^iP_t^{\max}\\ {}^iP_t^{\max}, & \text{if } {}^iP_t^{\text{ctrl}} > {}^iP_t^{\max}\end{cases}\end{aligned}$$

with

$$^iP_t^{\min} = \left[\left({}^iE_{\text{req}} - {}^ix_t\right)N_{\text{ctrl}} - \left({}^iT_{\text{dep}} - t - 1\right){}^iP_{\text{lim}}\right]_+ \tag{15}$$

$$^iP_t^{\max} = \min\left(\left({}^iE_{\text{req}} - {}^ix_t\right)N_{\text{ctrl}},\ {}^iP_{\text{lim}}\right) \tag{16}$$

with $^iP_t^{\min}$ the minimum power required to reach $^ix_t = {}^iE_{\text{req}}$ at time $^iT_{\text{dep}}$, and $^iP_t^{\max}$ the maximum power, limited by $^iP_{\text{lim}}$ and the charged energy ^ix_t. These constraints assure a valid charging power for the EVs.

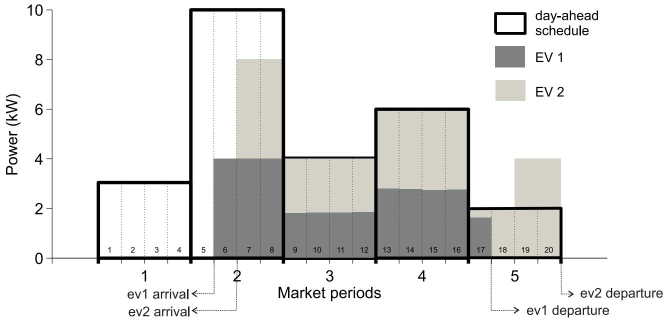

Fig. 3. Example 1: dispatch of a five hours day-ahead schedule between two EVs.

Algorithm 1 Priority-Based Dispatch f_{heur}

```
Input: $t, \mathbf{x}_t, \Omega_t, P_t^{\text{da}}$
I^sort ← based on $\Omega_t$, sort indices of the EVs by descending values of heuristic $^i\tau = (^iE_{\text{req}} - {}^ix_t)/((^iT_{\text{dep}} - t)\ {}^iP_{\text{lim}})$
for $i = \mathbf{I}_1^{\text{sort}}, \ldots, \mathbf{I}_{|\Omega_t|}^{\text{sort}}$ do
    if $P_t^{\text{da}} > 0$ then
        $^iP_t^{\text{ctrl}} = \frac{^i\tau}{\tau_{\text{tot}}}P_t^{\text{da}}$
        $^iP_t^{\text{ctrl}} = \min(\max(^iP_t^{\min},\ {}^iP_t^{\text{ctrl}}),\ {}^iP_t^{\max})$
        $P_t^{\text{da}} = P_t^{\text{da}} - {}^iP_t^{\text{ctrl}}$
    else
        $^iP_t^{\text{ctrl}} = 0$
    end if
end for
Output: $\{^iP_t^{\text{ctrl}} \mid i \in \mathbf{I}^{\text{sort}}\}$
```

The core of the dispatch performed by an aggregator (Fig. 2) is the dispatch algorithm (Algorithm 1). In function form

$$\mathbf{P}_{\text{t}}^{\text{ctrl}} = f_{\text{heur}}\left(t, \mathbf{x}_t, \Omega_t, P_t^{\text{da}}\right). \tag{17}$$

This algorithm takes the current time t, the charged energy $\mathbf{x}_t$, the EV parameters Ω_t in (10), and the day-ahead power P_t^{da} as input. Based on these inputs, the dispatch algorithm calculates a charging priority for each EV [22], which acts as a heuristic to divide power between the EVs based on their "urgency" to charge. For example, an EV with an empty battery will typically have a higher priority than an EV with a nearly full battery. The output of the algorithm is a control power $^iP_t^{\text{ctrl}}$ for each EV, which is communicated to the respective EV in step 3. Finally, the EVs calculate their actual charging power based on the policy π in (14).

Example 1: In Fig. 3, an example of the heuristic dispatch of a given day-ahead schedule between two EVs is shown. EV 1 requires 8 kWh and is available from control period 6 to 17. EV 2 requires 10 kWh and is available from control period 7 to 20. The maximum charging power of both EVs is 4 kW, and each market period contains 4 control periods. During each control period, the dispatch algorithm aims to minimize the difference between day-ahead schedule and total charging power of the EVs. In market period 1, the EVs are not available yet, so the input to the dispatch algorithm contains no EV information, which results in an empty set of control actions.

In market period 2, the EVs are charged at their maximum power as soon as they arrive. Nonetheless, this is not enough to obtain the requested power of 10 kW. In market period 3 and 4, the dispatch algorithm exactly follows the day-ahead schedule by dividing the scheduled power between EVs. Although EV 1 leaves earlier, EV 2 obtains a slightly higher power value, as its heuristic value is higher due to the large amount of energy still to be charged. In market period 5, the EVs depart, which leads to overcharging, because EV 2 did not charge its battery yet. As a result, a positive imbalance was observed in market period 1 and 2, no imbalance in market period 3 and 4, and a negative imbalance in market period 5.

As illustrated in example 1, the heuristic dispatch algorithm follows a predefined day-ahead schedule as good as possible. Nonetheless, when EVs are not available, do not have the required amount of power, or require immediate charging, the charging power will deviate from the day-ahead schedule. This deviation will cause costly imbalances for the aggregator. Therefore, an important part of our approach is defining a day-ahead schedule in decision phase I, which can be dispatched in decision phase II.

B. Learning Day-Ahead Schedule (Decision Phase I)

In the first decision phase (day D-1), the aggregator determines the day-ahead schedule to submit. This decision making process is formalized as an MDP with state space $S = X \times \Omega \times \{1, \ldots, T\}$ and action space P^{da}. In P^{da}, the control actions are a discretization of $[0, P_{\mathrm{grid}}]$, with P_{grid} the maximum allowed total charging power determined by the power grid. Control actions are only able to affect the state space component X [the EVs' charged energy in (11)]. The uncontrollable state space component Ω [the EVs' charging parameters in (10)] is determined by the random disturbances $\mathbf{w}_t = \{{}^i\theta_D | t = {}^iT_{\mathrm{arr}} - 1,\ {}^iT_{\mathrm{arr}} \in^i \theta_D\}$, which are characterized by a joint probability distribution $P_t(.)$. The state transitions of $\mathbf{x}_t \in X$ and $\Omega_t \in \Omega$ are described by

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \pi\left(t, \mathbf{x}_t, \Omega_\mathbf{t}, f_{\mathrm{heur}}\left(t, \mathbf{x}_t, \Omega_\mathbf{t}, P_t^{\mathrm{da}}\right)\right)\Delta t \quad (18)$$

$$\Omega_{\mathbf{t+1}} = \Omega_\mathbf{t} \cup \mathbf{w_t} \quad (19)$$

where the transition of $\mathbf{x}_t$ is defined by substituting (17) in (14), and (14) in (13). The cost signal in this MDP is obtained by substituting the state variables $\mathbf{x}_t$ and control action P_t^{da} in the aggregator cost function (5)[2]

$$c_t(\mathbf{x}_t, P_t^{\mathrm{da}}, \Omega_t) = P_t^{\mathrm{da}}\Delta t\lambda_t + \left[\Sigma\pi(.)\Delta t - P_t^{\mathrm{da}}\Delta t\right]_+ \lambda_t^- - \left[P_t^{\mathrm{da}}\Delta t - \Sigma\pi(.)\Delta t\right]_+ \lambda_t^+ \quad (20)$$

where λ_t, λ_t^-, and λ_t^+ are the prices defined in the respective market period h such that $t \in \mathbf{T}(h)$.

The goal of decision phase I is to find an optimal open-loop policy π_o^*, which selects for control time t the action $P_t^{\mathrm{da},*}$ based only on the initial state s_1 of the system ($P_t^{\mathrm{da},*} = \pi_o^*(t, s_1)$). The resulting actions define a full day-ahead schedule $\{P_1^{\mathrm{da},*}, \ldots, P_T^{\mathrm{da},*}\}$, which minimizes the expected T-stage cost

$$J^{\pi^*}(s_1) = \mathbb{E}\left(\sum_{t=1}^{T} c_t(\mathbf{x}_t, P_t^{\mathrm{da},*}, \Omega_t)\right). \quad (21)$$

Algorithm 2 Obtain $\mathbf{P}^{da}$ From $\hat{Q}$

```
Input: $\hat{Q}$, estimate for $z_1$.
for $t = 1, \ldots T$ do
    $P_t^{da} = \arg\min_{u'} \hat{Q}(t, z_t, u')$
    $z_{t+1} = z_t + P_t^{da}$
end for
return $\mathbf{P}^{da}$
```

For any practical amount of EVs (N $\geq$ 10) the curse of dimensionality quickly results in an intractable MDP. To alleviate this curse, a feature extraction is used [23], which maps the state $\mathbf{x}_t$ to z_t, defined as

$$z_t = \sum \mathbf{x}_t \quad (22)$$

which is the total charged energy of the EV fleet. Consequently, the new system state is described by (t, z_t, Ω_t). By extracting a feature, the control problem can be considered as a problem of imperfect state information [23]. In this paper, only the present values of z_t are used. However, these values can be readily extended with past information of the states visited and control actions selected.

Based on the feature z_t and substituting (6) and (7), the cost function in (20) can be written as

$$c_t\left(z_t, P_t^{\mathrm{da}}, z_{t+1}\right) = \Delta z_t\lambda_t + \beta\left|P_t^{\mathrm{da}}\Delta t - \Delta z_t\right| \quad (23)$$

where $\Delta z_t = z_{t+1} - z_t$ is the total energy charged by the EV fleet between time t and $t+1$. The first part of this formula ($\Delta z_t\lambda_t$) is the cost for buying the actual charged energy (during decision phase II) at the day-ahead market. The second part ($\beta\,|..|$) is the opportunity cost for buying too little (negative imbalance) or too much (positive imbalance) at the day-ahead market. Consequently, when $\beta > 0$, the opportunity cost of the optimal day-ahead schedule will be zero. This is an important result as it implies that for an optimal policy in a deterministic case

$$\left|P_t^{\mathrm{da}}\Delta t - \Delta z_t\right| = 0 \quad (24)$$

which is approximately true in a stochastic case. This property of an optimal policy will be exploited in our learning algorithm.

If all information describing the MDP would be known, $\mathbf{P}^{\mathrm{da}}$ in (8) can be obtained using a direct policy search algorithm [24]. However, because the disturbances and dynamics of the EVs are unknown during decision phase I, a batch RL approach is used, which learns from past experience. In this approach, a policy is improved each day, by observing the performance of the policy in preceding days or "episodes." Examples of candidate algorithms for calculating an open-loop policy are model-free Monte Carlo estimation and fitted Q iteration-policy evaluation (FQI-PE) [25] in combination

[2] For notational convenience, the parameters of function π in (18) are omitted.

with a generic optimizer such as cross entropy. However, in this paper, we efficiently calculate a policy based on the property in (24). This property allows to find $\mathbf{P}^{\text{da},*}$ by following the procedure presented in Algorithm 2, i.e., first calculating the optimal P_t^{da} for every energy state by using a $\hat{Q}$-function, and then retrieving $P^{\text{da},*}$ by stepping forward in time based on (24). As Ω is an uncontrollable state component, the averaged Q-function Q_Ω is used (for notational convenience, Q is used in stead of Q_Ω)

$$Q_\Omega(z,t) = \mathop{\mathbb{E}}_{\Omega}\{Q(z,t,\Omega)\}. \tag{25}$$

To calculate the $\hat{Q}$-function, FQI [18] is used (Algorithm 3). Based on information of previous episodes in a batch $\mathcal{F}$ of tuples $(z_t, P_t^{\text{da}}, z_{t+1}, c_t)$. In these tuple, z_t denotes the total energy charged at time t, P_t^{da} the day-ahead power control action at time t, z_{t+1} the successive total energy charged at time $t+1$, and c_t the cost as calculated in (23). Exploration in these episodes is achieved by using Boltzmann exploration [19], which for each time t selects an action P_t^{da} with probability

$$\mathbf{P}\left(z_t, P_t^{\text{da}}\right) = \frac{e^{Q_t(z_t,P_t^{\text{da}})/\tau_D}}{\sum_{P_t^{\text{da}}} e^{Q_t(z_t,P_t^{\text{da}})/\tau_D}}. \tag{26}$$

In this formula, Q_t is linearly scaled in the interval [0, 100]. A temperature $\tau_D = 100$ will select all actions with similar probability, while subsequent lower values $\tau_{D+1} = \tau_D - \Delta\tau$ will result in a greedy policy, which only selects higher valued actions.

V. Evaluation

In this section, the heuristic-based RL approach is benchmarked against multistage SP, which is able to calculate the optimal solution in a predefined EV fleet model. Although the *a priori* availability of an exact EV fleet model is unlikely, SP provides us with an upper bound on solution quality. The goal of this evaluation is to determine to which degree the RL approach can learn a day-ahead schedule, without using an EV fleet model.

A. EV Fleet Model

In order to define an SP benchmark, an artificial EV fleet model is defined. In this model, each EV is characterized by tuples of the form $(t_{\text{start}}, t_{\text{end}}, E_{\text{req}})$, wherein t_{start} is the begin time of charging, while t_{end} is the end time of charging. Within the interval $[t_{\text{start}} \ldots t_{\text{end}}]$, E_{req} has to be charged.

A daily scenario for the EV fleet is fully defined by one charging cycle per EV. A complete set of possible scenarios is defined as

$$\mathcal{S} \triangleq \left\{s_1, \ldots, s_{N_s}\right\} \tag{27}$$

for $n = 1, \ldots, N_s$

$$s_n = \left\{\left({}^it_{\text{start}}^n,\ {}^it_{\text{end}}^n,\ {}^iE_{\text{req}}^n\right), i = 1, \ldots, N_{\text{ev}}\right\}. \tag{28}$$

In this evaluation, we assume a small company which has a fleet of 15 EVs with mode 1 charging capabilities (charging power limited to 3.3 kW). Each EV is used in a different work shift, as in [26]. In the reference scenario (Table II), 4 EVs are used during the morning shift (∼6–14h), 8 EVs during the day shift (∼9–17h), and 3 EVs during the afternoon shift (∼12–20h). Each EV in a particular shift will arrive, depart and request an amount of energy according to an artificial probability distribution. This distribution was chosen to introduce sufficient variability to benchmark our solution, while limiting the number of scenarios to keep the required computational resources for SP within the capabilities of our workstation.[3]

TABLE II
Reference Scenario of EVs Charging at Work

Shift	nr of EVs	p	$t_{\text{arr}}(h)$	$t_{\text{dep}}(h)$	E_{req} (kWh)
Morning	4	0.5	6	14	11
		0.5	7	15	10
Day	8	0.4	8	17	15
		0.6	9	18	14
Afternoon	3	0.3	11	19	7
		0.7	12	20	6

Algorithm 3 Fitted Q Iteration [18, p. 508]

```
Input: a collection of four-tuples $\mathcal{F}$ and a regression algorithm.
Initialization:
Set n to $T$.
Let each $\hat{\mathcal{Q}}_t \in \{\hat{\mathcal{Q}}_t \mid t = 1, \ldots, T\}$ be a function equal to zero everywhere on $Z \times P^{\text{da}}$.
Iterations:
Repeat until $n = 1$
    - n $\leftarrow n - 1$
    - Build the training set $\mathcal{TS} = \{(i^l, o^l), l = 1, \ldots, \#\mathcal{F}\}$ based on the function $\hat{\mathcal{Q}}_{n+1}$ and on the set of four-tuples $\mathcal{F}$
          $i_t^l = (z_t^l, P_t^{\text{da},l})$,   (29)
          $o_t^l = c_t^l + \gamma \min_{u \in [0, P_{\text{grid}}]} \hat{\mathcal{Q}}_{n+1}(z_{t+1}^l, u)$.   (30)
    - Use the regression algorithm to induce from $\mathcal{TS}$ the function $\hat{\mathcal{Q}}_n(z_t, P_t^{\text{da}})$.
```

B. Benchmark: SP

To evaluate our approach in terms of optimality, a SP benchmark is defined. This benchmark uses the exact model of the EV fleet, as described in Section V-A. The complete stochastic optimization problem is

$$\min_{\mathbf{E}^{\text{da}},\mathbf{P}} \sum_{h\in\mathcal{H}} \lambda_h E_h^{\text{da}} \tag{31a}$$

$$+ \sum_{n=1}^{N_{\text{scen}}} \pi^n \sum_{h=1}^{H} \lambda_h^- \left[\sum_{t\in\mathbf{T}(h)} \sum_{i=1}^{N_{\text{ev}}} {}^iP_t^n \Delta t - E_h^{\text{da}}\right]_+ \tag{31b}$$

$$- \sum_{n=1}^{N_{\text{scen}}} \pi^n \sum_{h=1}^{H} \lambda_h^+ \left[E_h^{\text{da}} - \sum_{t\in\mathbf{T}(h)} \sum_{i=1}^{N_{\text{ev}}} {}^iP_t^n \Delta t\right]_+ \tag{31c}$$

[3] Intel Xeon processor (3.46 GHz, 12 MB cache, 4 cores) and 12 GB of RAM.

subject to

$$\forall n \in \{1, \ldots, N_\mathrm{s}\}, \forall t \in \{1, \ldots, T\} :$$
$$0 \leq \sum_{i=1}^{N_\mathrm{ev}} {}^{i}P_t^n \leq P_\mathrm{grid} \tag{32}$$

$$\forall n \in \{1, \ldots, N_\mathrm{s}\}, \forall i \in \{1, \ldots, N_\mathrm{ev}\} :$$
$$\sum_{t={}^{i}t^n_\mathrm{start}}^{{}^{i}t^n_\mathrm{end}} {}^{i}P_t^n \Delta t = {}^{i}E^n_\mathrm{req} \tag{33}$$

$$\forall n \in [1, \ldots, N_\mathrm{s}], \forall i \in [1, \ldots, N_\mathrm{ev}], \forall t \in \{1, \ldots, T\} :$$
$$0 \leq {}^{i}P_t^n \leq {}^{i}P_\mathrm{lim} \tag{34}$$

$$\forall n1, n2 \in [1, \ldots, N_\mathrm{s}], \forall i \in [1, \ldots, N_\mathrm{ev}] :$$
$${}^{i}P_t^{n1} = {}^{i}P_t^{n2} \quad \forall t \in \{t \mid \xi^{n1}_{[t]} = \xi^{n2}_{[t]}\}. \tag{35}$$

The objective of our optimization problem is minimizing the summation of three terms as can be seen in (31). In the first term (31a), the purchase costs in the day-ahead market are defined. In this term, the optimization variable is the day-ahead load schedule $\mathbf{E}^\mathrm{da}$. In the second term (31b), the sum of the expectation value of the negative imbalance costs associated with each scenario are defined. In the third term (31c), the sum of the expectation value of the positive imbalance costs associated with each scenario are defined. In both the second and third term, the optimization variable is the charging power of each EV during each control period.

Equations (32)–(35) define the four constraints of our optimization problem, which hold for each scenario s. In (32), the maximum collective EV power consumption is defined. We assume this constraint is put in place by the grid operator. In (33), the charging energy defined in each tuple is coupled to the charging power of each respective EV. In (34), the maximum individual EV power consumption is defined, determined by the power limitations of its local connection. In (35), the nonanticipativity constraints are defined. In this formula, $\xi^{n1}_{[t]} = \xi^{n2}_{[t]}$ is the "history equality," defined in the Appendix.

C. Simulation Results: Benchmarking the RL Approach

In this section, a series of four experiments is described and discussed. In each experiment, the RL solution is compared in different situations with the optimal solution. Day-ahead prices are used from the Belgian power exchange platform Belpex [21].

In the first experiment, the aggregator's cost progress in the reference scenario is analyzed. In Fig. 4, the mean and standard deviation of the costs observed in 100 independent simulation runs are shown. Each day in a simulation run, a different driving behavior is observed, based on the probability distribution of the EVs (Table II). Before the first day, the aggregator has no information about its fleet, and buys a steady amount of energy during the whole day. After 20∼30 days of exploration, the cost converges toward the optimal cost calculated by the benchmark. This optimal cost varies daily, depending on the day-ahead prices of the respective day. Two key parameters in this experiment are the temperature step in the Boltzmann exploration $\Delta\tau (= 5)$ and the "simultaneity factor" $\mathbf{f}_s (= 0.5)$. The influence of these parameters is analyzed in experiment 2 and 3.

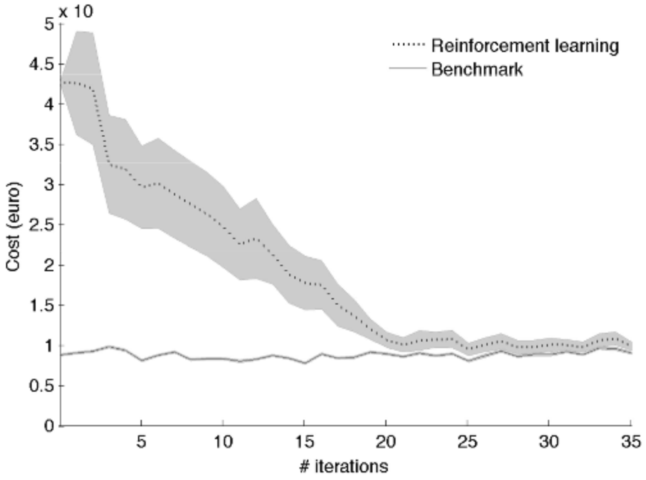

Fig. 4. Cost evolution of RL solution benchmarked in the reference scenario.

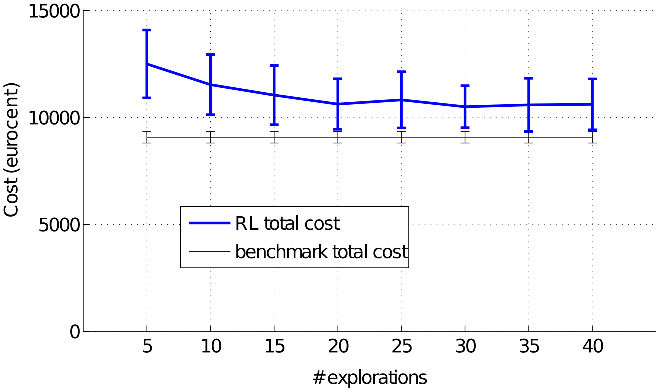

Fig. 5. Cost evolution of RL solution compared to benchmark.

In the second experiment, the influence of the temperature step $\Delta\tau$ of Boltzmann's exploration probability in (26) is analyzed. In Fig. 5, $\Delta\tau$ is varied from 20 (≈5 exploration steps) to 2.5 (≈40 exploration steps). For each value of these parameters, the result of 100 simulation runs are shown. In each simulation run, when the temperature reaches 0, the solution quality is recorded. From these results, we observe that the cost already converges after 20 iterations ($\Delta\tau = 5$). This fast convergence is achieved by using fitted Q iteration. Based on these results, a value of 5 is used for $\Delta\tau$.

In the third experiment, the influence of grid constraints is analyzed. While the aggregator wants to charge EVs at the lowest day-ahead prices, distribution grid constraints have to be taken into account. Typically, DSOs size their feeders and transformers based on an empirical value, called a simultaneity factor (sometimes called diversity factor). This factor expresses the expected peak load as a fraction of the maximum possible load. Based on the simultaneity factor,

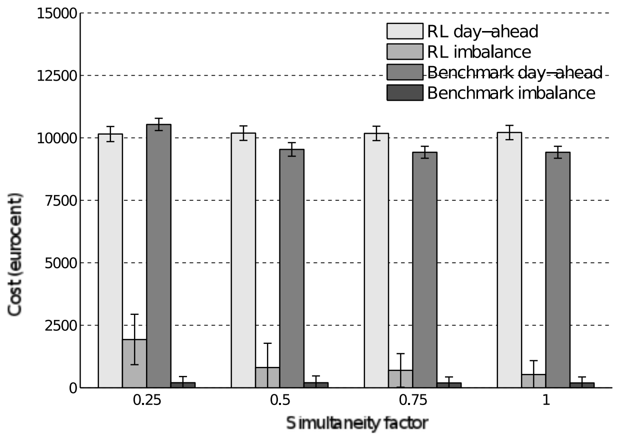

Fig. 6. Cost of RL solution benchmarked for different simultaneity factors.

a grid constraint can be calculated

$$P_{\text{grid}} = \mathbf{f}_s \sum_{i=1}^{N_{\text{ev}}} {}^{i}P_{\text{lim}}. \tag{36}$$

In Fig. 6, the costs are shown for a grid designed with simultaneity factor 0.25, 0.5, 0.75, and 1. The aggregator's costs are higher for a lower simultaneity factor, because these prevent EVs from charging during the lowest prices. Between the fitted Q iteration and stochastic benchmark, a similar difference in total costs (≈13%) is observed for each simultaneity factor. For the simultaneity factors 0.5, 0.75, and 1, the balance between imbalance and day-ahead costs is similar. However, for a simultaneity factor of 0.25, the RL approach has proportionally more imbalance costs and less day-ahead costs. The reason for this difference is that the problem is very constraint (the aggregated load of the benchmark solution looks like a block function), which limits the solution space. Consequently, our heuristic RL approach is forced to create imbalances, but compensates by lowering day-ahead costs. In case the simultaneity factor is smaller than 0.25, the EV charging problem becomes overconstrained, and some EVs will not be able to charge any more without overloading the grid. In this case, an additional cost for not charging EVs should be added to our cost function, which is out of scope of this paper.

In the fourth experiment, the solution quality in terms of the total "EV flexibility" is analyzed. In this paper, we define EV flexibility as the ability to shift an EVs charging energy in time. In general, longer charging times and less requested energy increase EV flexibility. In this experiment, we varied the EV flexibility by adding variation to our reference scenario (Table II). The requested energy for each car is now defined by a normal distribution $\mu = E_{\text{req}}$ and $\sigma = 2$, and the chance p for different arrival and departure times is now defined by a normal distribution $\mu = p$ and $\sigma = 1$. In Fig. 7, results are shown for 100 independent simulation runs, which shows an average cost increase of 10%.

In summary, all four experiments show that our approach is able to learn a cost-effective day-ahead schedule under varying circumstances, without using any *a priori* information about the EVs. The small-scale scenario used for this evaluation enabled us to calculate a benchmark solution. In the next section, our approach is simulated in a realistic large-scale environment.

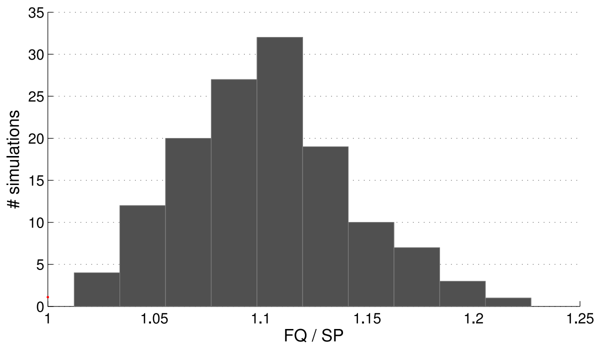

Fig. 7. Cost for a varying EV flexibility (FQ = fitted Q iteration cost, SP = SP cost).

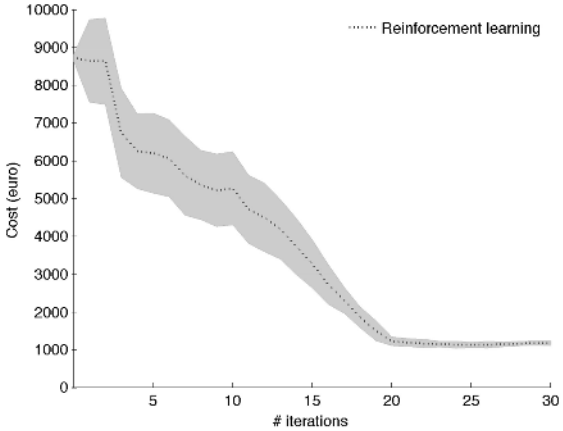

Fig. 8. Cost evolution of RL solution.

D. Simulation Results: Realistic Large-Scale Scenario

In this section, the performance of our RL approach is evaluated in a large scale, realistic scenario of EVs managed by an aggregator. The driving patterns of these EVs are based on statistical data on Belgian transport behavior [26]. Based on conventional vehicles, EV types are divided in subcompact, midsize and large vehicles, with each their specific power consumption (0.185, 0.220, and 0.293 kWh/km) and battery size (20, 30, and 40 kWh). Furthermore, each EV has a unique behavior for driving and parking (e.g., at home, work, or visits). We assume all EVs have mode 1 charging capabilities at each parking location, such that standard electrical plugs and outlets can be used. Consequently, the maximum electrical current per EV is 16A, which amounts to 3.3 kW (taken into account a maximum voltage drop of 10%). In Fig. 8, the mean and standard deviation of the costs in 100 independent simulations are shown for 2500 EVs. Similar as in experiment 1 in Section V-C, the cost converges after 20 iterations.

VI. Conclusion

In this paper, we studied an aggregator's problem of defining a day-ahead schedule to charge an EV fleet, in absence of an exact model of each EVs charging flexibility. On one hand, the aggregator wants to purchase its energy at low prices.

On the other hand, the aggregator wants to avoid imbalances, which cause high imbalance costs. To solve this problem, we proposed a RL approach to learn a cost-effective day-ahead consumption plan, which only uses readily available EV charging parameters. In a practical situation, an aggregator's choice between a model-based solution and a blind RL solution will depend on prior knowledge about the charging flexibility of an EV fleet, and the costs associated with constructing and maintaining a mathematical model.

Future and ongoing work focuses on learning heuristic demand of a heterogenous set of devices in different market environments. Examples of nonEV devices are heat pumps and electric boilers. Examples of different market environments are day-ahead markets where arbitration is allowed, intraday markets and ancillary service markets. In case of arbitration, artificial imbalance prices used in (5) will be substituted by predictions of imbalance prices. Furthermore, to provide incentives for consumers to provide EV charging flexibility, different pricing mechanism (see [27]) have to be compared.

Appendix
Definition of History Equality

History equality defines the conditions under which two scenarios cannot be distinguished. This concept is important for defining the nonanticipativity constraints in (35), which enforce the same control actions in scenarios with an equal history

$$\forall n \in [1, \ldots, N_{\mathrm{s}}], \forall t \in [1, \ldots, T] : \xi^n_{[t]} \triangleq (\xi^n_1, \xi^n_2, \ldots, \xi^n_t) \quad (37)$$

$$\xi^{n1}_{[t]} = \xi^{n2}_{[t]} \Leftrightarrow \forall s \in [1, \ldots, t] : \xi^{n1}_s = \xi^{n2}_s \quad (38)$$

$$\xi^{n1}_s = \xi^{n2}_s \Leftrightarrow \forall i \in [1, \ldots, N_{\mathrm{ev}}] : \left({}^i t^{n1}_{\mathrm{start}},\ {}^i t^{n1}_{\mathrm{end}},\ {}^i E^{n1}_{\mathrm{req}}\right) = \left({}^i t^{n2}_{\mathrm{start}},\ {}^i t^{n2}_{\mathrm{end}},\ {}^i E^{n2}_{\mathrm{req}}\right). \quad (39)$$

In (37), the history of a scenario n up to time t is defined as a sequence of uncertain data $\xi^n_1 \cdots \xi^n_t$ which is gradually revealed over time [28]. If this revelation of uncertain data is equal for two scenarios from 1 to t, these scenarios have an equal history at time t in (38). In (39), the uncertain data in our SP problem is defined: arrival time, departure time, and requested energy of an EV.

References

[1] S. Kamboj, W. Kempton, and K. S. Decker, "Deploying power grid-integrated electric vehicles as a multi-agent system," in *Proc. 10th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, vol. 1. Taipei, Taiwan, 2011, pp. 13–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=2030470.2030473

[2] R. J. Bessa and M. A. Matos, "The role of an aggregator agent for EV in the electricity market," in *Proc. 7th Mediterr. Conf. Exhibit. Power Gener. Transm. Distrib. Energy Convers. (MedPower)*, Agia Napa, Cyprus, 2010, pp. 1–9.

[3] J. Escudero-Garzas, A. Garcia-Armada, and G. Seco-Granados, "Fair design of plug-in electric vehicles aggregator for V2G regulation," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3406–3419, Oct. 2012.

[4] S. Mathieu, D. Ernst, and Q. Louveaux, "An efficient algorithm for the provision of a day-ahead modulation service by a load aggregator," in *Proc. 4th IEEE/PES Innov. Smart Grid Technol. Europe (ISGT EUROPE)*, Lyngby, Denmark, 2013, pp. 1–5.

[5] D. Wu, D. Aliprantis, and L. Ying, "Load scheduling and dispatch for aggregators of plug-in electric vehicles," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 368–376, Mar. 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/tsg/tsg3.html#WuAY12

[6] Siemens R&D and Microelectronics. (Jun. 2011). *Openv2g, Sourceforge*. [Online]. Available: http://openv2g.sourceforge.net

[7] OCPP. (2013). *Open Charge Point Protocol*. [Online]. Available: http://www.ocpp.nl/

[8] A. Al-Awami and E. Sortomme, "Coordinating vehicle-to-grid services with energy trading," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 453–462, Mar. 2012.

[9] M. Plazas, A. Conejo, and F. Prieto, "Multimarket optimal bidding for a power producer," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 2041–2050, Nov. 2005.

[10] M. Caramanis and J. Foster, "Coupling of day ahead and real-time power markets for energy and reserves incorporating local distribution network costs and congestion," in *Proc. 48th Annu. Allerton Conf. Commun. Control Comput.*, Allerton, IL, USA, 2010, pp. 42–49.

[11] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality* (Probability and Statistics). Hoboken, NJ, USA: Wiley, 2007.

[12] D. Lee and W. B. Powell, "An intelligent battery controller using bias-corrected Q-learning," in *Association for the Advancement of Artificial Intelligence (AAAI)*, J. Hoffmann and B. Selman, Eds. Palo Alto, CA, USA: AAAI Press, 2012.

[13] W. Shi and V. Wong, "Real-time vehicle-to-grid control algorithm under price uncertainty," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, 2011, pp. 261–266.

[14] M. Levorato, A. Goldsmith, and U. Mitra, "Residential demand response using reinforcement learning," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Gaithersburg, MD, USA, 2010, pp. 409–414.

[15] A. Rahimi-Kian, B. Sadeghi, and R. Thomas, "Q-learning based supplier-agents for electricity markets," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, vol. 1. San Francisco, CA, USA, 2005, pp. 420–427.

[16] M. Rahimiyan and H. Mashhadi, "An adaptive Q-learning algorithm developed for agent-based computational modeling of electricity market," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 5, pp. 547–556, Sep. 2010.

[17] P. P. Reddy and M. M. Veloso, "Strategy learning for autonomous agents in smart grid markets," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, Barcelona, Spain, 2011, pp. 1446–1451.

[18] D. Ernst, P. Geurts, L. Wehenkel, and L. Littman, "Tree-based batch mode reinforcement learning," *J. Mach. Learn. Res.*, vol. 6, pp. 503–556, Apr. 2005.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (Adaptive Computation and Machine Learning). London, U.K.: A Bradford Book, Mar. 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262193981

[20] A. Conejo, M. Plazas, R. Espinola, and A. Molina, "Day-ahead electricity price forecasting using the wavelet transform and ARIMA models," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1035–1042, May 2005.

[21] Belpex. (2011). *Belgian Power Exchange: Belpex*. [Online]. Available: http://www.belpex.be/index.php?id=78

[22] S. Vandael, B. Claessens, M. Hommelberg, T. Holvoet, and G. Deconinck, "A scalable three-step approach for demand side management of plug-in hybrid vehicles," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 720–728, Jun. 2013.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Belmont, MA, USA: Athena Scientific, 1996.

[24] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*, 1st ed. Boca Raton, FL, USA: CRC Press, 2010.

[25] R. Fonteneau, S. Murphy, L. Wehenkel, and D. Ernst, "Batch mode reinforcement learning based on the synthesis of artificial trajectories," *Ann. Oper. Res.*, vol. 208, no. 1, pp. 383–416, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10479-012-1248-5

[26] J. Van Roy *et al.*, "An availability analysis and energy consumption model for a Flemish fleet of electric vehicles," in *Proc. Eur. Elect. Veh. Congr. (EEVC)*, Brussels, Belgium, 2011, pp. 1–12. [Online]. Available: http://www.esat.kuleuven.be/electa/publications/fulltexts/pub_2221.pdf

[27] M. Kefayati and R. Baldick, "Energy delivery transaction pricing for flexible electrical loads," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, 2011, pp. 363–368.

[28] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia, PA, USA: Soc. Ind. Appl. Math. Math. Program. Soc., 2009. [Online]. Available: http://www2.isye.gatech.edu/people/faculty/Alex_Shapiro/SPbook.pdf

Stijn Vandael (M'14) was born in Zonhoven, Belgium. He received the master's degree in industrial engineering from Katholieke Hogeschool Limburg, Diepenbeek, Belgium, and the M.Sc. degree in computer science from Katholieke Universiteit Leuven, Leuven, Belgium, where he is currently pursuing the Ph.D. degree from the Department of Computer Science in close cooperation with the Department of Electrical Engineering.

His current research interests include coordination in multiagent systems, electric vehicles, and smart grids.

Bert Claessens was born in Neeroeteren, Belgium. He received the M.Sc. and Ph.D. degrees in applied physics from the University of Technology of Eindhoven, Eindhoven, The Netherlands, in 2002 and 2006, respectively.

In 2006, he was at ASML Veldhoven, Veldhoven, The Netherlands, as a Design Engineer. Since 2010, he has been a Researcher at the Vlaamse Instelling Voor Technologisch Onderzoek, Mol, Belgium. His current research interests include algorithm development and data analysis.

Damien Ernst (M'12) received the M.S. and Ph.D. degrees in engineering from the University of Liège, Liège, Belgium, in 1998 and 2003, respectively.

He is currently an Associate Professor with the University of Liège, where he is affiliated with the Systems and Modeling Research Unit. His current research interests include power system control and reinforcement learning.

Dr. Ernst is a EDF-Luminus Chair on smart grids.

Tom Holvoet (M'12) received the Ph.D. degree in computer science from Katholieke Universiteit Leuven, Leuven, Belgium, in 1997.

He is a Professor with the Department of Computer Science, Katholieke Universiteit Leuven. His current research interests include software engineering of decentralized and multiagent systems, software architecture, autonomic computing, and aspect-oriented software development.

Geert Deconinck (SM'00) received the M.Sc. degree in electrical engineering and the Ph.D. degree in applied sciences from KU Leuven, Leuven, Belgium, in 1991 and 1996, respectively.

From 1995 to 1997, he received a grant from the Flemish Institute for the Promotion of Scientific-Technological Research in Industry (IWT). He was a Post-Doctoral Fellow of the Fund for Scientific Research—Flanders (Belgium) (F.W.O.-V.) from 1997 to 2003. He has been a Full Professor with the University of Leuven, Leuven, since 2010, where he has been the Head of the Research Group Electrical Energy and Computing Architectures with the Department of Electrical Engineering, since 2012. He is the Scientific Leader for the research domain algorithms, modeling, optimization, applied to smart electrical, and thermal networks at Research Center EnergyVille, Genk, Belgium. His current research interests include dependable system architectures for automation and control, specifically in the context of smart electric distribution networks.

Prof. Deconinck is a fellow of the Institute of Engineering and Technology.