Achieving Scalable Model-Free Demand Response in Charging an Electric Vehicle Fleet with Reinforcement Learning

Nasrin Sadeghianpourhamami
IDLab, Ghent University-imec, Belgium
nasrin.sadeghianpourhamami@ugent.be

Johannes Deleu
IDLab, Ghent University-imec, Belgium
johannes.deleu@ugent.be

Chris Develder
IDLab, Ghent University-imec, Belgium
chris.develder@ugent.be

ABSTRACT

To achieve coordinated electric vehicle (EV) charging with demand response (DR), a model-free approach using reinforcement learning (RL) is an attractive proposition. Using RL, the DR algorithm is defined as a Markov decision process (MDP). Initial work in this area comprises algorithms to control just one EV at a time, because of scalability challenges when taking coupling between EVs into account. In this paper, we propose a novel MDP definition for charging an EV fleet. More specifically, we propose (1) a relatively compact aggregate state and action space representation, and (2) a batch RL algorithm (i.e., an instance of fitted Q-iteration, FQI) to learn the optimal EV charging policy.

ACM Reference Format:
Nasrin Sadeghianpourhamami, Johannes Deleu, and Chris Develder. 2018. Achieving Scalable Model-Free Demand Response in Charging an Electric Vehicle Fleet with Reinforcement Learning. In *e-Energy '18: International Conference on Future Energy Systems, June 12–15, 2018, Karlsruhe, Germany.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3208903.3212042

1 INTRODUCTION

Recently, reinforcement learning (RL) has been adopted for Demand response (DR) algorithms, to facilitate model-free control for coordinating the user flexibility. In RL based approaches, the DR problem is defined in the form of a Markov decision process (MDP). A coordinating agent interacts with the environment (i.e., DR participating customers, energy providers, energy market prices, etc.) and takes control actions while aiming to maximize the long term expected reward (or minimize the long term expected cost). The DR objective (e.g., load flattening, load balancing) is achieved by designing the reward/cost signal. One of the main challenges of RL based DR approaches is the curse of dimensionality due to the continuity and scalability of the state and the action spaces, which hinders the applicability of RL based DR for large scale problems.

The existing RL based DR solutions are either developed for an individual EV (e.g., [4] and [1]) or need a heuristic (which does not guarantee an optimum solution) to obtain EV fleet load during the learning process [5]. Indeed, a scalable MDP which generalizes to EV fleets with different characteristics (e.g., charging rates, fleet size) does not exist in current literature. In this paper we take the first step to fill this gap and propose a scalable MDP which takes into account both the individual EV charging characteristics (i.e., arrival time, charging and sojourn duration) as well as an aggregated fleet status. We do this by proposing a binning algorithm to define both the state and the action spaces in a compact yet informative fashion. Additionally, our proposed MDP can generalize to various feet sizes and charging rates. We then adopt batch reinforcement learning (fitted Q-iteration [2]) with function approximation to find the best EV charging policy.

e-Energy '18, June 12–15, 2018, Karlsruhe, Germany

ACM ISBN 978-1-4503-5767-8/18/06…$15.00
https://doi.org/10.1145/3208903.3212042

2 METHODOLOGY

The goal of the proposed EV charging is to minimize the long term cost of charging an EV fleet for an aggregator in a real-time decision-making scenario.

2.1 Markov Decision Process

We focus on the scenario of load flattening in this paper and regard this problem as a sequential decision making problem and formulate it using an MDP with unknown transition probability.

2.1.1 State Space. An EV charging session is characterized by: EV arrival time, time left till departure (Δt^{depart}), requested energy and EV charging rate. We translate the requested energy to time needed to complete the charging (Δt^{charge}). This implicitly assumes the same charging rates for all the EVs in a fleet. We will represent the EV fleet state in a 2D grid, with one axis representing Δt^{depart}, the other Δt^{charge}.

Let us define H_{max} as the maximum allowable connection time and Δs as the duration of the decision slot s, hence, $S_{max} = H_{max}/\Delta s$ is the maximum number of allowable connection slots. At each time slot s, an aggregate state of a fleet (i.e., $\mathbf{x}_s$) is obtained by binning the connecting EVs in set $V = \{(\Delta t_1^{depart}, \Delta t_1^{charge}), \ldots, (\Delta t_{N_s}^{depart}, \Delta t_{N_s}^{charge})\}$ into a 2D grid. Each tuple in V represents an EV. N_s is the number of connected EVs at time slot s, and takes a value between 0 and the fleet size N_{max}. Note that the size of $\mathbf{x}_s$ and hence the size of the state space is independent of the fleet size and is only influenced by S_{max} and Δs. This ensures scalability of the state space to various fleet sizes.

2.1.2 Action Space. The action at time slot s is a binary decision whether to charge (or not) the cars in each bin of the $\mathbf{x}_s$ matrix. Hence, a matrix of the same size as $\mathbf{x}_s$ is used to define the action at time slot s (i.e., $\mathbf{u}_s$). Each element of $\mathbf{u}_s$ is a binary number with

0 dictating to delay the EVs in the corresponding bins of $\mathbf{x}_s$ and 1 to charge them.

2.1.3 Cost function. The goal is to flatten the aggregate charging load of an EV fleet while ensuring the EVs charging is completed before departure. Hence, our cost function associated with each state action pair is the sum of two terms: (1) the cost of total fleet consumption for a decision slot($C^{demand}(\mathbf{x}_s, \mathbf{u}_s)$), and (2) the penalty for unfinished charging $C^{penalty}(\mathbf{x}_s, \mathbf{u}_s) = M \cdot I_{s+1}$, where I_{s+1} is the number of incomplete (or impossible to complete) charging (EVs with $\Delta t_n^{depart} < \Delta t_n^{charge}$) in the next state $\mathbf{x}_{s+1}$ as a consequence of taking action $\mathbf{u}_s$ at state $\mathbf{x}_s$. M is the penalty, which we set to be greater than $2^{N_{max}}$ to ensure the EVs charging is always completed before their departure (i.e., one incomplete EV is costlier than charging all EVs simultaneously).

2.1.4 System Dynamics. The system dynamics (environment) are defined using transition probabilities in the MDP framework: $P(\mathbf{x}_{s+1}|\mathbf{x}_s, \mathbf{u}_s)$. The transition probabilities are unknown in the EV fleet charging problem due to stochasticity of the EV arrivals and their charging demands.

2.2 State-Action Value function

Note that $C(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1})$ is the instantaneous cost an aggregator incurs when action $\mathbf{u}_s$ is taken at state $\mathbf{x}_s$ and leads to state $\mathbf{x}_{s+1}$. The objective is to find a control policy $\pi : \mathbf{X} \rightarrow \mathbf{U}$ that minimizes the expected T step cost, denoted by $Q^{\pi}(\mathbf{x}_1, \mathbf{u}_1)$, starting from state $\mathbf{x}_1$, and taking action $\mathbf{u}_1$. The optimum policy π^* satisfies the Bellman equation: $Q^*(\mathbf{x}_s, \mathbf{u}_s) = \min_{\mathbf{u}\in\mathbf{U}} \mathbb{E}\{C(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}) + Q^*(\mathbf{x}_{s+1}, \mathbf{u})\}$ However, solving the Bellman equation requires the knowledge of the transition probabilities, which are unknown in our problem. Hence, a learning algorithm should be used to obtain approximation $\widehat{Q}^*(\mathbf{x}, \mathbf{u})$. This can then be used to take control action $\mathbf{u}_s$, following: $\mathbf{u}_s = \text{argmin}_{u\in U}\, \widehat{Q}^*(\mathbf{x}_s, \mathbf{u})$

2.3 Batch Reinforcement Learning

Availability of EV charging datasets enables to adopt the batch mode RL algorithms to approximate $\widehat{Q}^*(\mathbf{x}, \mathbf{u})$ from past experience. Hence, we use the Fitted-Q-iteration (FQI) (depicted in Algorithm 1) to approximate $\widehat{Q}^*(\mathbf{x}, \mathbf{u})$. At its input, FQI takes a set of past experiences, $\mathcal{F}$, in form of tuples $(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}, C(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}))$. The state-action value function is initialized with zeros on the state-action space. In each iteration N of the algorithm, $Q_{N,s}$ is calculated for each tuple in $\mathcal{F}$ using the latest approximation of action-value function (Q_{N-1}) to form a labeled dataset $\mathcal{T}_{reg}$ to be used by function approximation to estimate Q_N.

3 AN EXAMPLARY SCENARIO

For illustrative purposes, we choose a simple scenario of charing 2 EVs within a $T = 3$ hour horizon. Lets assume that the duration of each decision slot is 1 hour and at time $t = 1$ we have 2 connecting cars: $V = \{(\Delta t_1^{depart}, \Delta t_1^{charge}) = (3, 2), (\Delta t_2^{depart}, \Delta t_2^{charge}) = (2, 1)\}$, with no other arrivals during the control horizon. Figure 1(a) illustrates the resulting state space using the binning algorithm at the first time slot. The resulting matrix is normalized by the fleet capacity (2 in this example). The shaded cells in the 2D grid of

Algorithm 1: Fitted Q-iteration using function approximation for estimating the T-step return

Input : $\mathcal{F} = \{(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}, C(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}))|s = 1, \ldots, |\mathcal{F}|\}$;

Initialize $\widehat{Q}_0$ to be zero everywhere on $\mathbf{X} \times \mathbf{U}$;

foreach $N = 1, \ldots, T$ **do**

  foreach $s = 1, \ldots, |\mathcal{F}|$ **do**

    $Q_{N,s}(\mathbf{x}_s, \mathbf{u}_s) \leftarrow C(\mathbf{x}_s, \mathbf{u}_s, \mathbf{x}_{s+1}) + \min_{\mathbf{u}\in\mathbf{U}} \widehat{Q}_{N-1}(\mathbf{x}_{s+1}, \mathbf{u})$

  Use function approximator to obtain $\widehat{Q}_N$ from $\mathcal{T}_{reg} = \{((\mathbf{x}_s, \mathbf{u}_s), Q_{N,s})|s = 1, \ldots, |\mathcal{F}|\}$

return $\widehat{Q}_T$

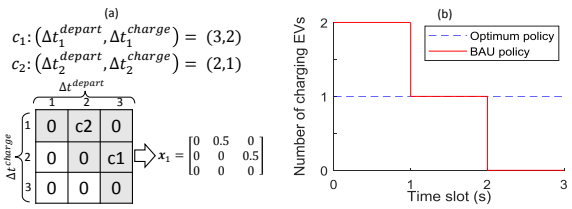

Fig. 1: An examplary scenario: (a) Binning 2 EVs (c_1 and c_2) to construct the aggregare state matrix $\mathbf{x}_1$ at $s = 1$ (b) Number of charging EVs for BAU vs. optimized charging policy

Fig. 1(a) indicate bins with $\Delta t^{charge} \leq \Delta t^{depart}$. EVs in these bins have enough time to complete their charging. However, once EVs fall into white cells in the 2D grid, it is no longer possible to complete their charging before their departure.

The optimum policy obtained from Algorithm 1 takes actions which flatten the load curve by avoiding the simultaneous charging at $s = 1$ and instead shifting the charging of one of the EVs from $s = 1$ to $s = 2$ as shown in Fig. 1(b).

4 FUTURE WORK

As a next step, we aim to use the proposed MDP framework to control the charging of larger real-world EV fleets with longer control horizon and longer allowable connected times and tackle the following challenges: (1) Long allowable connection times increase the size of the state matrix $\mathbf{x}_s$ and results in larger state space which might challenge the exploration. (2) Longer control horizon will require more iterations in the FQI algorithm, since FQI calculates the T-step return where T is proportional to the length of the control horizon, (3) With larger state space and longer horizons, the resulting space of possible state-action sequences becomes too large and impossible to be fully presented to the FQI as an exhaustive set $\mathcal{F}$. Hence, a tree exploration strategy should be used to sample the environment. (4) With bigger state and action matrices, as well as larger environment, more sophisticated neural networks should be examined for function approximation. (5) Extension to other DR use cases beyond load flattening will be explored. (6) The proposed methodology will be evaluated on the real-world EV charging dataset (e.g., [3]).

REFERENCES

[1] A. Chis, J. Lundén, and V. Koivunen. 2017. Reinforcement Learning-Based Plug-in Electric Vehicle Charging With Forecasted Price. *IEEE Transactions on Vehicular Technology* 66, 5 (May 2017), 3674–3684. https://doi.org/10.1109/TVT.2016.2603536

[2] Martin Riedmiller. 2005. Neural fitted Q iteration-first experiences with a data efficient neural reinforcement learning method. In *16th European Conference on Machine Learning*, Vol. 3720. Springer, 317–328.

[3] Nasrin Sadeghianpourhamami, Nazir Refa, Matthias Strobbe, and Chris Develder. 2018. Quantitive analysis of electric vehicle flexibility: A data-driven approach. *Int. J. Electr. Power Energy Syst.* 95 (Feb. 2018), 451–462. https://doi.org/10.1016/j.ijepes.2017.09.007

[4] Wenbo Shi and V. W. S. Wong. 2011. Real-time vehicle-to-grid control algorithm under price uncertainty. In *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 261–266. https://doi.org/10.1109/SmartGridComm.2011.6102330

[5] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck. 2015. Reinforcement Learning of Heuristic EV Fleet Charging in a Day-Ahead Electricity Market. *IEEE Transactions on Smart Grid* 6, 4 (July 2015), 1795–1805. https://doi.org/10.1109/TSG.2015.2393059