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:

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.

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 (Δdepart), requested energy and EV charging rate. We translate the requested energy to time needed to complete the charging (Δcharge). 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 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 fleet sizes.

The state of a fleet is defined as the number of connected EVs at time slot s, and takes a value between 0 and the maximum number of allowable connection slots. At each time slot s, an aggregate state of a fleet (i.e., x_s) is obtained by binning the connecting EVs in set V = {Δt_{depart}, Δt_{charge}, ..., Δt_{N_e}, Δt_{charge}} into a 2D grid. Each tuple in V represents an EV. N_e is the number of connected EVs at time slot t, and takes a value between 0 and the fleet size N_max. Note that the size of x_s and hence the size of the state space is independent of the fleet size and is only influenced by N_max and Δs. This ensures scalability of the state space to various fleet sizes.

2.1.2 Action Space. The action at time slot t is a binary decision whether to charge (or not) the cars in each bin of the x_s matrix. Hence, a matrix of the same size as x_s is used to define the action at time slot t (i.e., u_s). Each element of u_s is a binary number with...
0 dictating to delay the EVs in the corresponding bins of \( x_s \) 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 \( C_{\text{demand}}(x_s, u_t) \), and (2) the penalty for unfinished charging \( C_{\text{penalty}}(x_s, u_t) = M \cdot I_{s+1} \), where \( I_{s+1} \) is the number of incomplete (or impossible to complete) charging (EVs with \( \Delta t^{\text{depart}} < \Delta t^{\text{charge}} \)) in the next state \( x_{s+1} \) as a consequence of taking action \( u_t \) at state \( x_s \). \( M \) is the penalty, which we set to be greater than \( 2N_{\text{max}} \) 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: \( P(x_{s+1} | x_s, u_t) \). 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 \( C(x_s, u_t, x_{s+1}) \) is the instantaneous cost an aggregator incurs when action \( u_t \) is taken at state \( x_s \) and leads to state \( x_{s+1} \). The objective is to find a control policy \( \pi : X \rightarrow U \) that minimizes the expected \( T \) step cost, denoted by \( Q^\pi(x_t, u_t) \), starting from state \( x_t \) and taking action \( u_t \). The optimum policy \( \pi^* \) satisfies the Bellman equation: 

\[
Q^*(x_t, u_t) = \min_{u_t \in U} E[C(x_t, u_t, x_{s+1}) + Q^*(x_{s+1}, u)]
\]

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 \( \hat{Q}^*(x_t, u_t) \). This can then be used to take control action \( u_t \) following: 

\[
\hat{u}_t = \text{argmin}_{u_t \in U} \hat{Q}^*(x_t, u_t)
\]

2.3 Batch Reinforcement Learning

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

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

- **Input**: \( \mathcal{F} = \{(x_t, u_t, x_{s+1}, C(x_t, u_t, x_{s+1})) | s = 1, \ldots, |\mathcal{F}|\} \)
  1. Initialize \( \hat{Q}_0 \) to be zero everywhere on \( X \times U \)
  2. **foreach** \( N = 1, \ldots, T \) **do**
     3. **foreach** \( s = 1, \ldots, |\mathcal{F}| \) **do**
        4. \( \hat{Q}_{N,s}(x_t, u_t) \leftarrow C(x_t, u_t, x_{s+1}) + \min_{u_t \in U} \hat{Q}_{N-1}(x_{s+1}, u) \)
  5. Use function approximator to obtain \( \hat{Q}_N \)
  6. **return** \( \hat{Q}_T \)

Fig. 1(a) indicates bins with \( \Delta t^{\text{charge}} \leq \Delta t^{\text{depart}} \). 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 \( X \) 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


