An Online Learning Algorithm for Demand Response in Smart Grid

Shahab Bahrami, *Student Member, IEEE*, Vincent W.S. Wong, *Fellow, IEEE*, and Jianwei Huang, *Fellow, IEEE*

***Abstract*—Demand response program with real-time pricing can encourage electricity users towards scheduling their energy usage to off-peak hours. A user needs to schedule the energy usage of his appliances in an online manner since he may not know the energy prices and the demand of his appliances ahead of time. In this paper, we study the users' long-term load scheduling problem and model the changes of the price information and load demand as a Markov decision process, which enables us to capture the interactions among users as a partially observable stochastic game. To make the problem tractable, we approximate the users' optimal scheduling policy by the Markov perfect equilibrium (MPE) of a fully observable stochastic game with incomplete information. We develop an online load scheduling learning (LSL) algorithm based on the actor-critic method to determine the users' MPE policy. When compared with the benchmark of not performing demand response, simulation results show that the LSL algorithm can reduce the expected cost of users and the peak-to-average ratio (PAR) in the aggregate load by 28% and 13%, respectively. When compared with the short-term scheduling policies, the users with the long-term policies can reduce their expected cost by 17%.**

***Keywords*: Demand response, real-time pricing, partially observable stochastic game, online learning, actor-critic method.**

I. Introduction

The future smart grid aims to empower utility companies and users to make more informed energy management decisions. This motivates the utility companies to provide users with incentives to adjust the timing of their electricity usage [1]. The incentives may be through a demand response program with time-varying pricing schemes such as real-time pricing (RTP) and inclining block rate (IBR) pricing [2]. With a properly designed demand response program, the utility company can decrease its generation cost due to the reduction of peak-to-average ratio (PAR) in the aggregate load. Meanwhile, users can reduce their payment by taking advantage of low prices at off-peak hours.

There are several challenges for users to optimally determine their energy schedule in a demand response program. First, if the utility company uses RTP or IBR, the users' scheduling decisions are coupled since the appliances' energy schedule of a user affects the price that is charged to all users, hence affects other users' cost. Second, each user is uncertain about the total demand of other users, as well as the time of use and operation constraints of his own appliances. In particular, each appliance's operation depends on its task specifications (e.g., task duration, start time/deadline of the task), which are not known *a priori* until the user decides to turn on that appliance. Third, the users may not know the price information ahead of time.

There have been some efforts in tackling the above challenges. We divide the related literature into two main threads. The first thread is concerned with techniques for scheduling the energy usage of the appliances in a household with a *myopic* user, who aims to minimize his cost in a short period of time (e.g., one day). Samadi *et al.* in [3] proposed pricing algorithms based on stochastic approximation to minimize the PAR of the aggregate load in one day for a single household. Chen *et al.* in [4] proposed a robust optimization approach to minimize the worst-case daily bill payment of a myopic user in a market with the RTP scheme. Eksin *et al.* in [5] captured the interactions among myopic users with heterogeneous but correlated consumption preferences with the RTP scheme as a Bayesian game. Forouzandehmehr *et al.* in [6] proposed a differential stochastic game framework to capture the interactions among myopic users with controllable appliances. In these works, however, it was not mentioned how the proposed scheduling algorithms can be used for *foresighted* users, who aim to minimize their long-term costs.

The second thread is concerned with techniques for scheduling the appliances in a household with a foresighted user. Wen *et al.* in [7] proposed a reinforcement learning algorithm to address the appliances scheduling problem in a household. Kim *et al.* in [8] proposed a load scheduling algorithm based on Q-learning for a microgrid with time-of-use pricing scheme. Liang *et al.* in [9] proposed a Q-learning approach to minimize the bill payment and discomfort cost of a foresighted user in a household. Ruelens *et al.* in [10] proposed a batch reinforcement learning algorithm to schedule controllable loads such as water heater and heat-pump thermostat. These works, however, did not mention how the proposed learning algorithms can capture the decision making of multiple foresighted users. Xiao *et al.* in [11] applied dynamic programming to model the interactions among multiple foresighted suppliers. Yao *et al.* in [12] studied the electricity sharing problem among multiple foresighted users with the RTP scheme. The scheduling problem of each individual user is formulated as a Markov decision process. A specific structure for the suboptimal policy of each user is determined. Jia *et al.* in [13] proposed a

Manuscript received on Oct. 8, 2016, revised on Jan. 11, 2017, and accepted on Feb. 2, 2017. This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Strategic Project Grant (STPGP 447607-13), and the Theme-based Research Scheme (Project No. T23-407/13-N) from the Research Grants Council of the Hong Kong Special Administrative Region, China. S. Bahrami and V.W.S. Wong are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, V6T 1Z4. J. Huang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, email: {bahramis, vincentw}@ece.ubc.ca, jwhuang@ie.cuhk.edu.hk

learning algorithm based on stochastic approximation for the utility company to determine the day ahead price values in a market with multiple foresighted users. These works, however, did not study the operation constraints of different electrical appliances in residential sectors.

In this paper, we focus on designing a load scheduling learning (LSL) algorithm for multiple residential users, who schedule their appliances in response to RTP information. Each user is aware that the total energy consumption (not just his own) will affect the price announced by the utility company. Furthermore, each user is selfish and aims to minimize his own bill payment. We study the long-term interactions among foresighted users instead of the short-term interactions among myopic users. It enables us to model the users' decision making with uncertainty about the price information and load demand of their appliances as a *Markov decision process* with different states for different possible scenarios. We capture the interactions among users as a *stochastic game* [14]. Markov perfect equilibrium (MPE) is a standard solution concept for analyzing stochastic games. Several algorithms have been proposed to determine an MPE in *fully observable* stochastic games [15]–[22]. Some algorithms are model-based and require knowledge of the dynamics of the system, i.e., the state transition probabilities. The model-based learning algorithms include rational learning methods [15]–[17], linear programming based algorithms [18], [19], and homotopy method [20]. Some other learning algorithms are model-free and aim to determine an MPE when the system dynamics are unknown. Examples of model-free approaches include Lyapunov optimization [21] method and reinforcement learning algorithms [22]. In the demand response program, the underlying game is *partially observable* [23]–[25], since each user only observes his own state and is uncertain about other users' states. The key challenge in our model is to characterize the MPE under the partial observability of each user and the interdependency among the users' policies. This paper is an extension of our previous work [26] that takes into account the uncertainty in the energy price and users' load demand.

The contributions of this paper are as follows:

- *Novel Solution Approach*: The partially observable stochastic game is a realistic framework to model the interactions among users, but it is difficult to solve. To make the problem tractable, we propose an algorithm executed by each user to approximate the state of all users using some additional information from the utility company. It enables us to approximate the users' optimal policy by the MPE policy in a fully observable stochastic game with incomplete information, which is more tractable.
- *Learning Algorithm Design*: We formulate an individual optimization problem for each household, its global optimal solution corresponds to the MPE policy of the proposed fully observable stochastic game with incomplete information. We develop an actor-critic method [27]–[30]-based distributed LSL algorithm that converges to the MPE policy. The algorithm is online and model-free, which enables users to learn from the consequences of their past decisions and schedule their appliances in an online fashion without knowing the system dynamics.
- *Performance Evaluation*: We evaluate the performance of the LSL algorithm in reducing the PAR in the aggregate load and the expected cost of users. Compared with the benchmark of not performing demand response, our results show that the LSL algorithm can reduce the PAR in the aggregate load and the expected cost of foresighted users by 13% and 28%, receptively. We compare the policy of the foresighted and myopic users, and show that foresighted users can reduce their daily cost by 17%. When compared with the Q-learning method (e.g., in [7] and [8]), the LSL algorithm based on the actor-critic method converges faster to the MPE policy.

The rest of this paper is organized as follows. Section II introduces the system model. In Section III, we model the interactions among users as a partially observable stochastic game and approximate it by a fully observable stochastic game with incomplete information. In Section IV, we develop a distributed learning algorithm to compute the MPE. In Section V, we evaluate the performance of the proposed algorithm through simulations. Section VI concludes the paper.

II. System Model

We consider a system with one utility company and a set $\mathcal{N}=\{1,\ldots,N\}$ of N households. Each household is equipped with an energy consumption controller (ECC) responsible for scheduling the appliances in that household. The ECC is connected to the utility company via a two-way communication network, which enables the exchange of the price information and the household's load demand. Users participate in demand response program for a long period of time (e.g., several weeks). We divide the time into a set $\mathcal{T}=\{1,\ldots,T\}$ of T equal time slots, e.g., 15 minutes per time slot. In this paper, we use ECC, household, and user interchangeably.

A. Appliances Model

Let $\mathcal{A}_i = \{1,\ldots,A_i\}$ denote the set of appliances in household $i \in \mathcal{N}$, where A_i is the total number of appliances. In each time slot, an appliance is either *awake* or *asleep*, indicating whether it is ready to operate or not. We define the appliance's operation state as follows:

Definition 1 (Appliance Operation State): For household $i \in \mathcal{N}$, the operation state of appliance $a \in \mathcal{A}_i$ in time slot $t \in \mathcal{T}$ is a tuple $\boldsymbol{s}_{a,i,t} = (r_{a,i,t},\, q_{a,i,t},\, \delta_{a,i,t})$, where $r_{a,i,t}$ is the number of remaining time slots to complete the current task, $q_{a,i,t}$ is the number of time slots for which the current task can be delayed, and $\delta_{a,i,t}$ is the number of time slots since the most recent time slot that appliance a becomes awake with the most recent new task.

Fig. 1 shows the values of $r_{a,i,t}$, $q_{a,i,t}$, and $\delta_{a,i,t}$ for appliance $a \in \mathcal{A}_i$, which has a task that should be operated for three time slots with a maximum delay of three time slots. When appliance a becomes awake in time slot t, $r_{a,i,t}$ and $q_{a,i,t}$ are initialized based on the current task (e.g., here we have $r_{a,i,t} = q_{a,i,t} = 3$), and $\delta_{a,i,t}$ is set to 1. The value of $r_{a,i,t}$ decreases when appliance a executes its task and becomes 0 when the appliance has completed its task and is

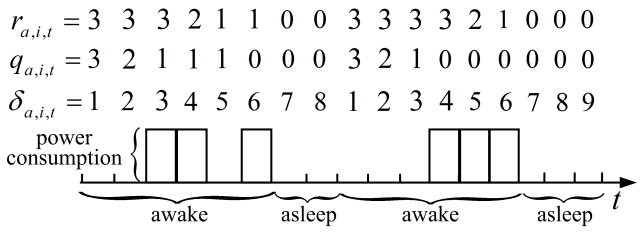


Fig. 1. The values of $r_{a,i,t}$, $q_{a,i,t}$, and $\delta_{a,i,t}$ for appliance a, which should be operated for three time slots with a maximum delay of three time slots.

asleep in time slot t. The value of $q_{a,i,t}$ remains unchanged when the task is executed, and decreases when the task is delayed. When $q_{a,i,t}$ is 0, the ECC cannot delay the appliance's task. The value of $\delta_{a,i,t}$ increases in each time slot and is reset to 1 when appliance a becomes awake with a *new* task. The appliance may start a new task right after completing the current task. Thus, without becoming asleep, $r_{a,i,t}$ and $q_{a,i,t}$ are initialized based on the new task, and $\delta_{a,i,t}$ is set to 1.

ECC i does not know when an appliance becomes awake ahead of time. Instead, it has a belief regarding $P_{a,i}(\delta_{a,i,t})$, the probability that the difference between two sequential wake-up times for appliance a is $\delta_{a,i,t}$, for $\delta_{a,i,t} \geq 1$. Such a probability distribution can be estimated, for example, based on the awake history for appliance a. ECC i can approximate $P_{a,i}(\delta_{a,i,t})$ by the ratio of the events that the difference between two consecutive wake-up times is $\delta_{a,i,t}$ in a given historical data record. Appliance a may become awake in the next time slot (for a new task) if either appliance a is asleep or it will complete the current task in the current time slot. In Appendix A, we show that given current time t, the probability $P_{a,i,t+1}$ that appliance $a \in \mathcal{A}_i$ becomes awake with a new task in the next time slot $t+1 \in \mathcal{T}$ is

$$P_{a,i,t+1} = \frac{P_{a,i}(\delta_{a,i,t})}{1 - \sum_{\Delta=1}^{\delta_{a,i,t}-1} P_{a,i}(\Delta)}. \tag{1}$$

We partition the set of appliances into *must-run* and *controllable*. Let $\mathcal{A}_i^{\mathrm{M}}$ denote the set of must-run appliances in household i. Examples of must-run appliances include lighting and TV. The ECC has no control over the operation of must-run appliances. On the other hand, the ECC can control the time of use for the controllable appliances. The set of controllable appliances in household i can further be partitioned into two sets: the set $\mathcal{A}_i^{\mathrm{N}}$ of *non-interruptible* appliances, and the set $\mathcal{A}_i^{\mathrm{I}}$ of *interruptible* appliances. Examples of non-interruptible appliances include washing machine and dish washer, and examples of interruptible appliances include air conditioner and electric vehicle (EV). The ECC may schedule a non-interruptible appliance during several consecutive time slots, but cannot interrupt its task. The ECC may delay or interrupt the operation of an interruptible appliance.

Each time an appliance $a \in \mathcal{A}_i$ becomes awake, it sends information about its new *task's specifications* to the ECC i.

Definition 2 (Task's Specifications): For an appliance $a \in \mathcal{A}_i$, the specifications of its task include the average power consumption $p_{a,i}^{\mathrm{avg}}$ to execute the task, the scheduling window $\mathcal{T}_{a,i} = [t_{a,i}^{\mathrm{s}}, t_{a,i}^{\mathrm{d}}]$ corresponding to a time interval which includes the earliest start time $t_{a,i}^{\mathrm{s}} \in \mathcal{T}$ and the deadline $t_{a,i}^{\mathrm{d}} \in \mathcal{T}$ for the task, the operation duration $d_{a,i}$ for a must-run or non-interruptible appliance corresponding to the total number of time slots required to complete the task, and the interval $[d_{a,i}^{\min}, d_{a,i}^{\max}]$ for an interruptible appliance corresponding to the range of the operation duration.

The value of the average power consumption $p_{a,i}^{\mathrm{avg}}$ is assumed to be fixed and known *a priori* for each appliance a. The operation duration $d_{a,i}$ for a non-interruptible appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ is fixed. On the other hand, the operation duration $d_{a,i}$ for a task of an interruptible appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ can be any value in the range of $[d_{a,i}^{\min}, d_{a,i}^{\max}]$, and we have $d_{a,i}^{\min} \geq 0$ and $d_{a,i}^{\max} \leq t_{a,i}^{\mathrm{d}} - t_{a,i}^{\mathrm{s}}$.

We use the binary decision variable $x_{a,i,t} \in \{0,1\}$ to indicate whether an appliance $a \in \mathcal{A}_i$ is scheduled to operate in time slot t ($x_{a,i,t} = 1$) or not ($x_{a,i,t} = 0$). Notice that $x_{a,i,t}$ is equal to 0 when appliance a is asleep (i.e., $r_{a,i,t} = 0$). Let $\boldsymbol{x}_{i,t} = (x_{a,i,t},\, a \in \mathcal{A}_i)$ denote the scheduling decision vector for all appliances in household i in time slot t.

ECC i can infer the state $\boldsymbol{s}_{a,i,t+1}$ of appliance a in the next time slot $t+1$ from the current state $\boldsymbol{s}_{a,i,t}$, the probability $P_{a,i,t+1}$, appliance's type, the task's specifications, and the scheduling decision $x_{a,i,t}$ as follows:

1) Must-run appliances: The feasible action for appliance $a \in \mathcal{A}_i^{\mathrm{M}}$ in time slot $t \in \mathcal{T}$ is

$$x_{a,i,t} = \begin{cases} 1, & \text{if } r_{a,i,t} \geq 1 \\ 0, & \text{if } r_{a,i,t} = 0. \end{cases} \tag{2}$$

When appliance $a \in \mathcal{A}_i^{\mathrm{M}}$ becomes awake with a new task, $r_{a,i,t}$ is set to $d_{a,i}$, and ECC i operates the appliance without delay, i.e., $q_{a,i,t}$ is equal to 0. Given current time t, the operation state in time slot $t+1$ can be obtained as follows:

- If either appliance $a \in \mathcal{A}_i^{\mathrm{M}}$ is asleep (i.e., $r_{a,i,t} = 0$) or it will complete its task in the current time slot (i.e., $r_{a,i,t} = 1$), then appliance a becomes awake in time slot $t+1$ with probability $P_{a,i,t+1}$, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (d_{a,i},\, 0,\, 1), \tag{3}$$

  and the appliance is asleep in time slot $t+1$ with probability $1 - P_{a,i,t+1}$, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (0,\, 0,\, \delta_{a,i,t} + 1). \tag{4}$$

- If $r_{a,i,t} \geq 2$, then appliance $a \in \mathcal{A}_i^{\mathrm{M}}$ has not completed its task yet. With probability 1, the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (r_{a,i,t} - 1,\, 0,\, \delta_{a,i,t} + 1). \tag{5}$$

2) Non-interruptible controllable appliances: The feasible action for appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ in time slot $t \in \mathcal{T}$ is

$$x_{a,i,t} = \begin{cases} 0 \text{ or } 1, & \text{if } t \in \mathcal{T}_{a,i},\ r_{a,i,t} \geq 1,\ q_{a,i,t} \geq 1, \\ 1, & \text{if } t \in \mathcal{T}_{a,i},\ r_{a,i,t} \geq 1,\ q_{a,i,t} = 0, \\ 0, & \text{if } r_{a,i,t} = 0. \end{cases} \tag{6}$$

Equation (6) implies that ECC i can decide to operate a non-interruptible appliance a or not when the appliance is awake

($r_{a,i,t} \geq 1$) and its current task can be delayed ($q_{a,i,t} \geq 1$). ECC i has to operate an awake appliance if the task cannot be delayed ($q_{a,i,t}=0$). ECC i will not schedule appliance a if it is asleep ($r_{a,i,t}=0$).

When appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ becomes awake, $r_{a,i,t}$ and $q_{a,i,t}$ are set to $d_{a,i}$ and $t_{a,i}^{\mathrm{d}} - t_{a,i}^{\mathrm{s}} - d_{a,i} + 1$, respectively. Given current time t, the operation state in the next time slot is as follows:

- If either appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ is asleep (i.e., $r_{a,i,t}=0$) or it will complete the current task in the current time slot (i.e., $r_{a,i,t}=1$ and $x_{a,i,t}=1$), then the appliance becomes awake in time slot $t+1$ with probability $P_{a,i,t+1}$, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (d_{a,i},\, t_{a,i}^{\mathrm{d}} - t_{a,i}^{\mathrm{s}} - d_{a,i} + 1,\, 1), \tag{7}$$

  and the appliance is asleep in time slot $t+1$ with probability $1-P_{a,i,t+1}$, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (0,\, 0,\, \delta_{a,i,t} + 1). \tag{8}$$

- If $r_{a,i,t} \geq 2$ and $x_{a,i,t}=1$, then appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ has not completed its task yet and is scheduled in the current time slot t. The appliance cannot be delayed in the next time slot, i.e., $q_{a,i,t+1}=0$. With probability 1, the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (r_{a,i,t} - 1,\, 0,\, \delta_{a,i,t} + 1). \tag{9}$$

- If $r_{a,i,t} \geq 1$ and $x_{a,i,t}=0$, then appliance $a \in \mathcal{A}_i^{\mathrm{N}}$ has not completed its task yet and is not scheduled in the current time slot t. With probability 1, we have $\boldsymbol{s}_{a,i,t+1} = (r_{a,i,t},\, q_{a,i,t}-1,\, \delta_{a,i,t}+1)$. The action set in (6) implies that $x_{a,i,t}$ cannot be equal to 0 if $q_{a,i,t}$ is 0 in time slot t.

3) Interruptible controllable appliances: Equation (6) is the feasible action for appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ in time slot $t \in \mathcal{T}$. When an interruptible appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ becomes awake with a new task, $r_{a,i,t}$ is set to the maximum operation duration $d_{a,i}^{\max}$. To operate the appliance for at least $d_{a,i}^{\min}$ time slots, ECC i can delay the task in at most $t_{a,i}^{\mathrm{d}} - t_{a,i}^{\mathrm{s}} - d_{a,i}^{\min} + 1$ time slots. The maximum operation duration may not be completed before the deadline within the scheduling horizon $\mathcal{T}_{a,i}$. In this case, if $t+1 \notin \mathcal{T}_{a,i}$, the interruptible appliance will become either asleep or awake with a new task in the next time slot $t+1$. The operation state in the next time slot $t+1$ is as follows:

- If the next time slot is not in the scheduling window (i.e., $t+1 \notin \mathcal{T}_{a,i}$), appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ is asleep (i.e., $r_{a,i,t}=0$), or the appliance will complete its task in the current time slot (i.e., $r_{a,i,t}=1$ and $x_{a,i,t}=1$), then the appliance becomes awake in time slot $t+1$ with probability $P_{a,i,t+1}$, with the next state as

$$\boldsymbol{s}_{a,i,t+1} = (d_{a,i}^{\max},\, t_{a,i}^{\mathrm{d}} - t_{a,i}^{\mathrm{s}} - d_{a,i}^{\min} + 1,\, 1), \tag{10}$$

  and the appliance is asleep in time slot $t+1$ with probability $1 - P_{a,i,t+1}$, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (0,\, 0,\, \delta_{a,i,t} + 1). \tag{11}$$

- If the next time slot is in the scheduling window (i.e., $t+1 \in \mathcal{T}_{a,i}$), $r_{a,i,t} \geq 2$, and $x_{a,i,t}=1$, then appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ is scheduled in the current time slot t. The appliance is awake in the next time slot $t+1$ with probability 1, and the next state is

$$\boldsymbol{s}_{a,i,t+1} = (r_{a,i,t} - 1,\, q_{a,i,t},\, \delta_{a,i,t} + 1). \tag{12}$$

- If $t+1 \in \mathcal{T}_{a,i}$, $r_{a,i,t} \geq 1$, and $x_{a,i,t}=0$, then the task of appliance $a \in \mathcal{A}_i^{\mathrm{I}}$ is not scheduled in the current time slot t. The appliance is awake in the next time slot $t+1$ with probability 1, with the corresponding next state as

$$\boldsymbol{s}_{a,i,t+1} = (r_{a,i,t},\, q_{a,i,t} - 1,\, \delta_{a,i,t} + 1). \tag{13}$$

B. Pricing Scheme and Household's Cost

In a dynamic pricing scheme, the payment by each household depends on the time and total amount of energy consumption. Let $l_{i,t} = \sum_{a \in \mathcal{A}_i} p_{a,i}^{\mathrm{avg}}\, x_{a,i,t}$ denote the aggregate load of household i in time slot t. Let l_t^{others} denote the aggregate background load demand of other users in time slot t that do not participate in the demand response program. The utility company knows l_t^{others} at the end of time slot t. Let $l_t = l_t^{\mathrm{others}} + \sum_{i \in \mathcal{N}} l_{i,t}$ denote the aggregate load demands of all users in time slot t.

We assume that the utility company uses a combination of RTP and IBR [3], [31]. In time slot $t \in \mathcal{T}$, the unit price λ_t is

$$\lambda_t(l_t) = \begin{cases} \lambda_{1,t}, & \text{if } 0 \leq l_t \leq l_t^{\mathrm{th}}, \\ \lambda_{2,t}, & \text{if } l_t > l_t^{\mathrm{th}}, \end{cases} \tag{14}$$

where $\lambda_{1,t} \leq \lambda_{2,t}$, $t \in \mathcal{T}$. Here, $\lambda_{1,t}$ and $\lambda_{2,t}$ are the unit price values in time slot t when the aggregate load is lower and higher than the threshold l_t^{th}, respectively. We define the vector of price parameters in time slot t as $\boldsymbol{\lambda}_t = (\lambda_{1,t}, \lambda_{2,t}, l_t^{\mathrm{th}})$. The price parameters are set by the utility company according to different factors such as the time of the day, day of the week, wholesale market conditions, and the operation conditions of the power network. We can capture the price changes by making the following assumption:

Assumption 1 The price parameters are generated according to a hidden Markov model.

In each hidden state, the price parameters are generated from a probability distribution which is unknown to the users [32], [33]. Assumption 1 is consistent with many realistic situations of price determination. For example, the price parameters $\boldsymbol{\lambda}_t$ may change periodically. In this case, the hidden states correspond to the time of the day, and the price parameters vector for each hidden state is fixed. In a more general model, a hidden state corresponds to the time of the day and the price parameters are chosen from a known probability distribution (e.g., a truncated normal distribution) in each hidden state. If this is the case, the probability distribution for each time slot can be estimated by examining the historical prices of the same time slot from many days [33]. In Section V, we compare the users scheduling decisions when the utility company applies the periodic and random price parameters, respectively.

The payment of household i in time slot t is $l_{i,t}\, \lambda_t(l_t)$. When the ECC interrupts the operation of the interruptible appliances, the corresponding user will experience a discomfort

cost. When an interruptible appliance $a \in \mathcal{A}^{\mathrm{I}}$ becomes awake, it sends the user's desirable operation schedule $x^{\mathrm{des}}_{a,i,t}$ for all time slots $t \in \mathcal{T}_{a,i}$ and the coefficients $\omega_{a,i,t}$, $a \in \mathcal{A}^{\mathrm{I}}_i$, $t \in \mathcal{T}_{a,i}$ (measured in terms of \$) to the ECC to reflect the user's discomfort caused by any potential change of the operation schedule of interruptible appliance a. For each household i, we capture the discomfort cost from scheduling the interruptible appliances by the weighted Euclidean distance between the operation schedule with demand response and the desirable operation schedule as $\sum_{a\in\mathcal{A}^{\mathrm{I}}_i} \omega_{a,i,t} \left| x_{a,i,t} - x^{\mathrm{des}}_{a,i,t} \right|$, which is also used in [34].

The total cost for each household i in time slot t involves the payment and discomfort cost. That is,

$$c_{i,t}(l_t) = l_{i,t}\, \lambda_t(l_t) + \sum_{a\in\mathcal{A}^{\mathrm{I}}_i} \omega_{a,i,t} \left| x_{a,i,t} - x^{\mathrm{des}}_{a,i,t} \right|. \tag{15}$$

In the long-term scheduling problem, the scheduling horizon T is a large number (e.g., if the scheduling horizon is six months and each time slot is 15 minutes, then we have $T \approx 17000$). Thus, it is reasonable to approximate the problem with an *infinite* scheduling horizon, and consider the expected discounted cost of each household i with the discount factor β [35, pp. 150] as

$$(1-\beta) \sum_{t=1}^{\infty} \beta^{t-1} c_{i,t}(l_{i,t}, \boldsymbol{l}_{-i,t}). \tag{16}$$

The parameter β in (16) can be used to characterize a wide range of users' behaviour. When β is close to zero, the users are myopic, i.e., they aim to minimize their short-term cost (e.g., daily cost) without considering the consequences of their short-term policy on their future cost. When β is close to one, the users are foresighted, i.e., they aim to minimize their long-term cost. One may assume different values of β for different participating users. In this paper, we assume that all users have the same value of β. In a more general future study, one may consider the case where different users have different values of β.

In the cost model (16) with an infinite scheduling horizon, we can consider the stationary scheduling decision making that is independent of time. Specifically, the decision making only depends on the price parameters and the appliance operation state in a time slot, but is independent of time slot index t. Therefore, we can remove time index t from the appliances' states, price parameters, and the household's cost.

III. Problem Formulation

Due to privacy concerns, each household does not reveal the information about its appliances to other households. We have

Assumption 2 The ECC can only observe the operation state of the appliances in its own household.

We capture the interactions among households in demand response program as a partially observable stochastic game.

Game 1 *Households' Partially Observable Stochastic Game*:
Players: The set of households $\mathcal{N}$.
States: The state of household i is $\boldsymbol{s}_i = (\boldsymbol{s}_{a,i},\, a \in \mathcal{A}_i)$.

Observations: The observation of household i is $\boldsymbol{o}_i = (\boldsymbol{s}_i, \boldsymbol{\lambda}) \in \mathcal{O}_i$, where $\mathcal{O}_i$ is the set of possible observations for household i. Let $\boldsymbol{o} = (\boldsymbol{o}_i,\, i \in \mathcal{N}) \in \mathcal{O}$ denote the observation profile of all households, where $\mathcal{O} = \prod_{i\in\mathcal{N}} \mathcal{O}_i$. We use notations $z(\boldsymbol{o}_i)$ and $z(\boldsymbol{o})$ to denote the value of an arbitrary parameter z in observation $\boldsymbol{o}_i$ of household i and observation profile of all households $\boldsymbol{o}$, respectively.

Actions: We define the action vector of household i in observation profile $\boldsymbol{o}$ as $\boldsymbol{x}_i(\boldsymbol{o}) = (x_{a,i}(\boldsymbol{o}),\, a \in \mathcal{A}_i)$. Let $\boldsymbol{x}(\boldsymbol{o}) = (\boldsymbol{x}_i(\boldsymbol{o}),\, i \in \mathcal{N})$ denote the action profile of all households. Let $\mathcal{X}_i(\boldsymbol{o}_i)$ denote the feasible action space obtained from (2), (6) for household i with observation $\boldsymbol{o}_i$.

Transition Probabilities: Given the current price parameters, Assumption 1 implies that the price parameters vector is Markovian. From Section II-A, the next state of an appliance depends only on its current state and action. Thus, the transition between the observations of a household is Markovian. Let $P_i(\boldsymbol{o}'_i \,|\, \boldsymbol{o}_i,\, \boldsymbol{x}_i(\boldsymbol{o}))$ denote the transition probability from observation $\boldsymbol{o}_i \in O_i$ to $\boldsymbol{o}'_i \in O_i$ with action $\boldsymbol{x}_i(\boldsymbol{o})$. It depends on the appliances wake-up probability in (1). Furthermore, the users have independent preferred plans of using their appliances. Hence, the states of different households are independent. The transition probability from observation $\boldsymbol{o} \in O$ to $\boldsymbol{o}' \in O$ with action profile $\boldsymbol{x}(\boldsymbol{o})$ is $P(\boldsymbol{o}' \,|\, \boldsymbol{o},\, \boldsymbol{x}(\boldsymbol{o})) = \prod_{i\in\mathcal{N}} P_i(\boldsymbol{o}'_i \,|\, \boldsymbol{o}_i,\, \boldsymbol{x}_i(\boldsymbol{o}))$.

Stationary Policies: Let $\pi_i(\boldsymbol{o}, \boldsymbol{x}_i(\boldsymbol{o}))$ denote the probability of choosing a feasible action $\boldsymbol{x}_i(\boldsymbol{o})$ in observation $\boldsymbol{o}$. Let $\boldsymbol{\pi}_i(\boldsymbol{o}) = (\pi_i(\boldsymbol{o}, \boldsymbol{x}_i(\boldsymbol{o})),\, \boldsymbol{x}_i(\boldsymbol{o}) \in \mathcal{X}_i(\boldsymbol{o}_i))$ denote the probability distribution over the feasible actions. We define the stationary policy for household i as the vector $\boldsymbol{\pi}_i = (\boldsymbol{\pi}_i(\boldsymbol{o}),\, \boldsymbol{o} \in \mathcal{O})$. Let $\boldsymbol{\pi} = (\boldsymbol{\pi}_i,\, i \in \mathcal{N})$ denote the joint policy of all households, and $\boldsymbol{\pi}_{-i}$ denote the policy for all households except household i.

Value functions: Under a given joint policy $\boldsymbol{\pi}$, the value function $V_i^{\boldsymbol{\pi}} : \mathcal{O} \to \mathbb{R}$ returns the expected discounted cost for household i starting with observation profile $\boldsymbol{o}$. It can be expressed as the following Bellman equation [14]:

$$V_i^{\boldsymbol{\pi}}(\boldsymbol{o}) = \mathbb{E}_{\boldsymbol{\pi}_i(\boldsymbol{o})} \left\{ Q_i^{\boldsymbol{\pi}_{-i}}\left(\boldsymbol{o}, \boldsymbol{x}_i(\boldsymbol{o})\right) \right\}, \qquad \forall\, \boldsymbol{o} \in \mathcal{O}, \tag{17}$$

where $\mathbb{E}_{\boldsymbol{\pi}_i(\boldsymbol{o})}\{\cdot\}$ denotes the expectation over the probability distribution $\boldsymbol{\pi}_i(\boldsymbol{o})$. Function $Q_i^{\boldsymbol{\pi}_{-i}}\left(\boldsymbol{o}, \boldsymbol{x}_i(\boldsymbol{o})\right)$ is the Q-function for household i with action $\boldsymbol{x}_i(\boldsymbol{o})$ in observation profile $\boldsymbol{o}$ when other households' policy is $\boldsymbol{\pi}_{-i}$ [14]. We have

$$Q_i^{\boldsymbol{\pi}_{-i}}\left(\boldsymbol{o}, \boldsymbol{x}_i(\boldsymbol{o})\right) = \mathbb{E}_{\boldsymbol{\pi}_{-i}(\boldsymbol{o})} \Big\{ (1-\beta)\, c_i\left(\boldsymbol{o}, \boldsymbol{x}(\boldsymbol{o})\right) + \beta \sum_{\boldsymbol{o}'\in\mathcal{O}} P\left(\boldsymbol{o}' \,|\, \boldsymbol{o}, \boldsymbol{x}(\boldsymbol{o})\right)\, V_i^{\boldsymbol{\pi}}(\boldsymbol{o}') \Big\}. \tag{18}$$

It is computationally difficult to determine the optimal policies for the households in such a partially observable stochastic game. In a partially observable stochastic game among users, each user needs to know what other users are observing in each time slot. Inspired by the works in [23]–[25], we propose an algorithm executed by each ECC to estimate the observation profile of all households. It enables us to study the users' optimal policy in a fully observable stochastic game with incomplete information, in which the households play a sequence of Bayesian games.

Algorithm 1 Executed by ECC $i \in \mathcal{N}$.

1: Communicate the average load demand $l_i^{\text{avg}}(\boldsymbol{o}_i)$ for all feasible actions $\boldsymbol{x}_i(\boldsymbol{o}_i) \in \mathcal{X}_i(\boldsymbol{o}_i)$ to the utility company.
2: Receive the average aggregate load $l^{\text{avg}}(\boldsymbol{o})$ from utility company.
3: Approximate the observation profile by $\hat{\boldsymbol{o}} := (l^{\text{avg}}(\boldsymbol{o}), \boldsymbol{\lambda})$.

A. Observation Profile Approximation Algorithm

To make the analysis of Game 1 tractable, we propose an algorithm executed by each ECC to approximate the observation of all households using some additional information. Let $\hat{\boldsymbol{o}}$ denote the approximate observation profile of all households. Algorithm 1 describes how ECC i obtains $\hat{\boldsymbol{o}}$. ECC i sends the average load demand $l_i^{\text{avg}}(\boldsymbol{o}_i)$ of all feasible actions $\boldsymbol{x}_i(\boldsymbol{o}_i) \in \mathcal{X}_i(\boldsymbol{o}_i)$ to the utility company. ECC i knows $\boldsymbol{\lambda}$ and receives the average aggregate load $l^{\text{avg}}(\boldsymbol{o}) = \frac{1}{N}\sum_{j\in\mathcal{N}} l_j^{\text{avg}}(\boldsymbol{o}_j)$. It approximates the observation profile $\boldsymbol{o}$ by vector $\hat{\boldsymbol{o}} = (l^{\text{avg}}(\boldsymbol{o}), \boldsymbol{\lambda})$.

In Algorithm 1, each household receives information on the average aggregate load demands. Thus, the privacy of each individual household is protected. All ECCs obtain the same approximation for an observation profile. Thus, we can consider a fully observable stochastic game with incomplete information. Under a given approximate observation profile $\hat{\boldsymbol{o}}$, the households play a Bayesian game, as each household i may have different observations $\boldsymbol{o}_i$, and thus different sets of feasible actions.

Game 2 *Households' Fully Observable Stochastic Game with Incomplete Information*:
This game is constructed from Game 1 if the households define their actions and policy as follows:

Actions: Let $\mathcal{O}_i(\hat{\boldsymbol{o}}) \subseteq \mathcal{O}_i$ denote the set of possible observations for household i in the approximate observation profile $\hat{\boldsymbol{o}}$. We define the set of actions for household i in the approximate observation profile $\hat{\boldsymbol{o}}$ as $\hat{\mathcal{X}}_i(\hat{\boldsymbol{o}}) = \{\boldsymbol{x}_i(\boldsymbol{o}_i) : \boldsymbol{x}_i(\boldsymbol{o}_i) \in \mathcal{X}_i(\boldsymbol{o}_i),\, \boldsymbol{o}_i \in \mathcal{O}_i(\hat{\boldsymbol{o}})\}$. The feasibility of an action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$ depends on the observation $\boldsymbol{o}_i$ of household i.

Policies: We define the stationary policy $\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\boldsymbol{o}_i))$ as the probability of choosing a *feasible* action $\boldsymbol{x}_i(\boldsymbol{o}_i) \in \mathcal{X}_i(\boldsymbol{o}_i)$ in an approximate observation profile $\hat{\boldsymbol{o}}$ when the observation of household i is $\boldsymbol{o}_i \in \mathcal{O}_i(\hat{\boldsymbol{o}})$. Let $P_i(\boldsymbol{o}_i|\hat{\boldsymbol{o}})$ be the probability that household i has observation $\boldsymbol{o}_i \in \mathcal{O}_i(\hat{\boldsymbol{o}})$ when the approximate observation profile is $\hat{\boldsymbol{o}}$. Hence, the probability of choosing any action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$ is $\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) = P_i(\boldsymbol{o}_i|\hat{\boldsymbol{o}})\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\boldsymbol{o}_i))$. Let $\boldsymbol{\pi}_i(\hat{\boldsymbol{o}}) = (\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})), \boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}}))$ denote the probability distribution over the actions for household i in an approximate observation profile $\hat{\boldsymbol{o}}$. We define the policy for household i in Game 2 as the vector $\boldsymbol{\pi}_i = (\boldsymbol{\pi}_i(\hat{\boldsymbol{o}}),\, \hat{\boldsymbol{o}} \in \mathcal{O})$.

B. Markov Perfect Equilibrium (MPE) Policy

In this subsection, we discuss how each household i determines a policy $\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})$ in Game 2 for any approximate observation profiles $\hat{\boldsymbol{o}}$ to minimize its value function $V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}})$. The MPE is a standard solution concept for the partially observable stochastic games. The MPE corresponds to the users' policies with Markov properties and is compatible with the assumption for the appliance model in Section II-A. The MPE in Game 2 is defined as follows:

Definition 3 A policy $\boldsymbol{\pi}^{\text{MPE}} = (\boldsymbol{\pi}_i^{\text{MPE}},\, i \in \mathcal{N})$ is an MPE if for every household $i \in \mathcal{N}$ with a policy $\boldsymbol{\pi}_i$, we have

$$V_i^{\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}^{\text{MPE}}\right)}(\hat{\boldsymbol{o}}) \geq V_i^{\left(\boldsymbol{\pi}_i^{\text{MPE}}, \boldsymbol{\pi}_{-i}^{\text{MPE}}\right)}(\hat{\boldsymbol{o}}), \quad \forall\, i \in \mathcal{N},\ \forall\, \hat{\boldsymbol{o}} \in \mathcal{O}. \quad (19)$$

The MPE policy is the fixed point solution of every household's *best response policy*. Household i solves the following Bellman equations when other households' policies are fixed:

$$V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}}) = \underset{\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})}{\text{minimize}}\ \mathbb{E}_{\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})}\left\{Q_i^{\boldsymbol{\pi}_{-i}^{\text{MPE}}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\right\},\ \forall\, \hat{\boldsymbol{o}} \in \mathcal{O}. \quad (20)$$

As the following Theorem states, the existence of the MPE is guaranteed for Game 2 .

Theorem 1 *Game 2 has at least one MPE in stochastic stationary policies.*

The proof of Theorem 1 can be found in Appendix B. The MPE is the fixed point of N recursive problems in (20) for all households. Problem (20) implies that for household i with action $\boldsymbol{x}_i(\hat{\boldsymbol{o}})$ under observation profile $\hat{\boldsymbol{o}}$ in the MPE, we have $V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}}) \leq Q_i^{\boldsymbol{\pi}_{-i}^{\text{MPE}}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$. We introduce an equivalent non-recursive optimization problem for each household, which is more tractable. For household $i \in \mathcal{N}$, we define the Bellman error [14] for an action $\boldsymbol{x}_i(\hat{\boldsymbol{o}})$ in an approximate observation profile $\hat{\boldsymbol{o}}$ as

$$B_i\left(V_i^{\boldsymbol{\pi}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right) = Q_i^{\boldsymbol{\pi}_{-i}}\left(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right) - V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}}). \quad (21)$$

We define function $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i)$ as the sum of the expected Bellman errors for all observations $\hat{\boldsymbol{o}} \in \mathcal{O}$. That is

$$f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i) = \sum_{\hat{\boldsymbol{o}}\in\mathcal{O}} \mathbb{E}_{\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})}\Big\{B_i\left(V_i^{\boldsymbol{\pi}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right)\Big\}. \quad (22)$$

Each household i aims to determine the policy $\boldsymbol{\pi}_i$ and the value function $V_i^{\boldsymbol{\pi}}$ to minimize $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i)$ by solving the following optimization problem.

$$\underset{V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i}{\text{minimize}}\ f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i) \quad (23)$$

$$\text{subject to } B_i(V_i^{\boldsymbol{\pi}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) \geq 0,\ \forall\, \hat{\boldsymbol{o}} \in \mathcal{O},\ \forall\, \boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}}).$$

Problem (23) is generally a non-convex problem, and may have several local minima. We show that the MPE policy of household i is the *global* minimum of problem (23).

Theorem 2 *The policy $\boldsymbol{\pi}^{\text{MPE}}$ is an MPE of Game 2 if and only if for all households $i \in \mathcal{N}$ with action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$, we have*

$$\pi_i^{\text{MPE}}\left(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right)\ B_i\big(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\big) = 0, \quad \forall\, \hat{\boldsymbol{o}} \in \mathcal{O}. \quad (24)$$

The proof can be found in Appendix C. Theorem 2 implies that the Bellman error is zero for an action with positive probability at the MPE. Thus, $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = 0$ and the MPE is the global optimal solution of problem (23) for all households.

Solving problem (23) is still challenging, as each ECC requires the values of the unavailable transition probabilities between the observations. This motivates us to develop a *model-free learning* algorithm that enables each ECC to sched-

ule the appliances in an online manner without knowing the system dynamics. Basically, each ECC updates the policy and value function based on the consequences of its past decisions.

As part of the learning algorithm, we need to record the observation and action spaces for a household. In order to reduce the complexity, we use the linear function approximation to estimate the value function [36, Ch. 3]. For household i, let $\boldsymbol{\phi}_i(\hat{\boldsymbol{o}}) = (\phi_{v,i}(\hat{\boldsymbol{o}}), v \in \mathcal{V})$ denote the row vector of basis functions, where $\mathcal{V}$ is the set of basis functions. Let $\boldsymbol{\theta}_i = (\theta_{v,i}, u \in \mathcal{V})$ denote the row vector of weight coefficients. The approximate value function for household i is

$$V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}}, \boldsymbol{\theta}_i) = \boldsymbol{\theta}_i \, \boldsymbol{\phi}_i^{\mathrm{T}}(\hat{\boldsymbol{o}}), \tag{25}$$

where T is the transpose operator. It enables ECC i to compute vector $\boldsymbol{\theta}_i$ with $|\mathcal{V}|$ elements instead of the value function $V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}})$ for all approximate observation profiles $\hat{\boldsymbol{o}}$. We parameterize the policy $\boldsymbol{\pi}_i$ for household i via *softmax approximation* [36, Ch. 3]. Let $\boldsymbol{\mu}_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) = \big(\mu_{p,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})), p \in \mathcal{P}\big)$ denote the row vector of basis functions, where $\mathcal{P}$ is the set of basis functions. Let $\boldsymbol{\vartheta}_i = (\vartheta_{p,i}, p \in \mathcal{P})$ denote the row vector of weight coefficients. The approximate probability of choosing action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$ is

$$\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}), \boldsymbol{\vartheta}_i) = \frac{e^{(\boldsymbol{\vartheta}_i \boldsymbol{\mu}_i^{\mathrm{T}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})))}}{\sum_{\boldsymbol{x}_i'(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})} e^{(\boldsymbol{\vartheta}_i \boldsymbol{\mu}_i^{\mathrm{T}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i'(\hat{\boldsymbol{o}})))}}. \tag{26}$$

To simplify the computation of this approximation, we use the vector of *compatible* basis functions $\boldsymbol{\psi}_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) = \big(\psi_{p,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})), p \in \mathcal{P}\big)$, where

$$\psi_{p,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) = \frac{\partial \ln(\pi_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}), \boldsymbol{\vartheta}_i))}{\partial \, \vartheta_{p,i}}. \tag{27}$$

We can show that for the softmax parameterized policy, the vector of basis functions $\boldsymbol{\mu}_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ can be replaced with vector $\boldsymbol{\psi}_i(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ [30].

IV. Online Learning Algorithm Design

In this section, we propose a load scheduling learning (LSL) algorithm executed by the ECC of each household to determine the MPE policy. We use an actor-critic learning method, which is more robust than the actor-only methods (such as the policy evaluation [22, Ch. 2]) and faster than the critic-only methods (such as the Q-learning and temporal difference (TD) learning [22, Ch. 6]). The concept of the actor-critic was originally introduced by Witten in [27] and then elaborated by Barto *et al.* in [28]. A detailed study of the actor-critic algorithm can be found in [29], [30]. Our LSL algorithm is based on the first proposed algorithm in [30]. The ECC is responsible for the actor and critic updates. In the critic update, the ECC evaluates the policy to update the value function. In the actor update, it updates the policy to decrease the objective value of problem (23) based on the updated value function. In the policy update, we use the gradient method with a smaller step size compared with the step size in the value function's update, thereby using a *two-timescale* update process [30].

Algorithm 2 describes the LSL algorithm executed by ECC i. The index k refers to both iteration and time slot. Our algorithm involves the initiation and scheduling phases. Line 1 describes the initialization in time slot $k = 1$. The loop involving Lines 2 to 14 describes the scheduling phase, which includes the observation profile approximation, the critic update, the actor update, and the basis function construction.

In Lines 3, ECC i executes Algorithm 1 to obtain the approximate observation profile $\hat{\boldsymbol{o}}$. In time slot $k = 1$, ECC i does not have any experience from its past decisions and chooses an action in Line 11. For $k > 1$, the critic and actor updates are executed. ECC i determines the updated vector $\boldsymbol{\theta}_i^k$ using the TD approach [22, Ch. 6]. The TD error e_{TD}^{k-1} is

$$\begin{aligned} e_{\mathrm{TD}}^{k-1} = (1-\beta) c_i \left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}^{k-1}(\hat{\boldsymbol{o}}^{k-1})\right) + \beta V_i^{\boldsymbol{\pi},k-1}\left(\hat{\boldsymbol{o}}_i^k, \boldsymbol{\theta}_i^{k-1}\right) \\ - V_i^{\boldsymbol{\pi},k-1}\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{\theta}_i^{k-1}\right). \end{aligned} \tag{28}$$

The critic update for ECC i is

$$\boldsymbol{\theta}_i^k = \boldsymbol{\theta}_i^{k-1} + \gamma_c^{k-1} \, e_{\mathrm{TD}}^{k-1} \boldsymbol{\phi}_i\big(\hat{\boldsymbol{o}}^{k-1}\big), \tag{29}$$

where γ_c^k is the critic step size in iteration k. In the actor update module, ECC i determines the updated vector $\boldsymbol{\vartheta}_i^k$ using the gradient method with descent direction. In particular, ECC i uses the descent direction $\pi_i^{k-1}\big(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{\vartheta}_i^{k-1}\big) \nabla_{\boldsymbol{\vartheta}_i^{k-1}} f_i^{\mathrm{obj}}\big(V_i^{\boldsymbol{\pi},k-1}, \boldsymbol{\pi}_i^{k-1}\big)$ to ensure convergence to the MPE. Since the gradient is not available, ECC i uses vector $e_{\mathrm{TD}}^{k-1} \boldsymbol{\psi}_i(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1}))$ as an estimate of the gradient [30, Algorithm 1]. Therefore, the convergence to the MPE is guaranteed, since the TD error e_{TD}^{k-1} is an estimate for the Bellman error for action $\boldsymbol{x}_i^{k-1}$ in iteration $k-1$. Thus, the descent direction is zero if condition (24) is satisfied. The actor update for ECC i is

$$\begin{aligned} \boldsymbol{\vartheta}_i^k = \boldsymbol{\vartheta}_i^{k-1} - \gamma_a^k \, \pi_i^{k-1}\big(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{\vartheta}_i^{k-1}\big) \\ \times e_{\mathrm{TD}}^{k-1} \, \boldsymbol{\psi}_i(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1})), \end{aligned} \tag{30}$$

where γ_a^k is the actor step size in iteration k. We use the approach in [37] to autonomously construct the new basis functions $\psi_{|\mathcal{P}|+1,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ and $\phi_{|\mathcal{V}|+1,i}(\hat{\boldsymbol{o}})$. The candidate for the basis function $\psi_{|\mathcal{P}|+1,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ is the TD error e_{TD}^{k-1} in (28), which estimates the Bellman error. The expectation over the Bellman errors of the feasible actions $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$ is the candidate for $\phi_{|\mathcal{V}|+1,i}(\hat{\boldsymbol{o}})$. We have

$$\psi_{|\mathcal{P}|+1,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) = e_{\mathrm{TD}}^{k-1}, \tag{31}$$

$$\phi_{|\mathcal{V}|+1,i}(\hat{\boldsymbol{o}}) = \mathbb{E}\left\{B_i^{k-1}\big(V_i^{\boldsymbol{\pi},k-1}, \hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\big)\right\}. \tag{32}$$

The expectation in (32) is over the probability of choosing each feasible actions $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. In Appendix D, we explain how to approximate the Bellman error for each feasible action. In Line 7, ECC i checks the convergence of $\boldsymbol{\theta}_i^k$ and decides whether to add the new basis functions or not.

In Line 11, ECC i schedules the appliances in the current time slot k. In Line 12, ECC i receives the cost $c_i\left(\hat{\boldsymbol{o}}_i^k, \boldsymbol{x}^k(\hat{\boldsymbol{o}}_i^k)\right)$. Next time slot is started in Line 13. In Line 14, the stopping criterion is given. From Theorem 2, LSL algorithm converges to the MPE if the objective value $f_i^{\mathrm{obj}}(V_i^{\boldsymbol{\pi},k-1}, \boldsymbol{\pi}_i^{k-1})$ is zero. ECC i computes the approximate objective value by summing over the expected Bellman errors up to iteration $k-1$ as

$$\widehat{f}_i^{\mathrm{obj}}(V_i^{\boldsymbol{\pi},k-1}, \boldsymbol{\pi}_i^{k-1}) =$$

Algorithm 2 LSL Algorithm Executed by ECC $i \in \mathcal{N}$.

1: Set $k := 1$, $\epsilon := 10^{-3}$, and $\xi = 10^{-3}$. Set $\phi_{1,i}(\cdot) := 1$ and $\psi_{1,i}(\cdot) := 1$, and randomly initialize $\theta_{1,i}^1$ and $\vartheta_{1,i}^1$.
2: **Repeat**
3: Observe $o_i^k := (\boldsymbol{s}_i^k, \boldsymbol{\lambda}^k)$. Approximate $\hat{\boldsymbol{o}}_i^k$ using Algorithm 1.
4: **If** $k \neq 1$,
5: Determine the updated vector $\boldsymbol{\theta}_i^k$ according to (29).
6: Determine the updated vector $\boldsymbol{\vartheta}_i^k$ according to (30).
7: **If** $|\boldsymbol{\theta}_i^k - \boldsymbol{\theta}_i^{k-1}| < \epsilon$,
8: Construct new basis functions $\psi_{|\mathcal{P}|+1,i}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ and $\phi_{|\mathcal{V}|+1,i}(\hat{\boldsymbol{o}})$ using (31) and (32).
9: **End if**
10: **End if**
11: Choose action $\boldsymbol{x}_i^k(\hat{\boldsymbol{o}}_i^k)$ using policy $\pi_i^k(\hat{\boldsymbol{o}}_i^k, \boldsymbol{\vartheta}_i^k)$.
12: Receive the cost $c_i\left(\hat{\boldsymbol{o}}_i^k, \boldsymbol{x}^k(\hat{\boldsymbol{o}}_i^k)\right)$ from the utility company.
13: $k := k + 1$.
14: **Until** $||\hat{f}_i^{\text{obj}}(V_i^{\boldsymbol{\pi}, k-1}, \boldsymbol{\pi}_i^{k-1})|| < \xi$.

$$\sum_{j=1}^{k-1} \mathbb{E}_{\boldsymbol{\pi}_i^{k-1}(\hat{\boldsymbol{o}}^j)} \left\{ B_i^j \left(V_i^{\boldsymbol{\pi}, k-1}, \hat{\boldsymbol{o}}^j, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^j) \right) \right\}. \tag{33}$$

The sufficient conditions for the actor and critic step sizes to ensure the convergence of the LSL algorithm are given in [29].

In the proposed model-free LSL algorithm, ECC i does not know the next states of the appliances until the next time slot begins in Line 13. The ECC updates its value function using the TD error in (28), which depends on the next time slot observation. Therefore, the ECC only goes through one iteration per time slot.

V. Performance Evaluation

In this section, we evaluate the performance of the LSL algorithm in a system, where one utility company serves 200 households that participate in the demand response program. The scheduling horizon is six months. Each time slot is 15 minutes. We consider six controllable appliances for each household, e.g., dish washer, washing machine, and stove are non-interruptible appliances, and EV, air conditioner, and water heater are interruptible appliances. We model other appliances such as refrigerator and TV as must-run appliances. Table I summarizes the task specifications of the controllable appliances [38]. For the EV in each household i, we have $d_{a,i}^{\min} = (B_i^{\text{d}} - B_i^0)/p_{a,i}^{\text{avg}}$ and $d_{a,i}^{\max} = (B_i^{\max} - B_i^0)/p_{a,i}^{\text{avg}}$, where B_i^0 is the initial charging level when the EV awakes, B_i^{d} is the charging demand for the next trip, and $B_i^{\max}$ is the battery's maximum capacity. The charging demand of the EV in household i is uniformly chosen at random from the set $\{18\,\text{kWh}, 18.75\,\text{kWh}, \ldots, 24\,\text{kWh}\}$. The battery capacity is set to 30 kWh. Typically, the user is indifference between the charging patterns for the EV as long as the charging is finished before the deadline. Thus, we set coefficients $\omega_{a,i,t}$, $t \in \mathcal{T}$ to zero for the EV. Coefficients $\omega_{a,i,t}$, $t \in \mathcal{T}$ are chosen uniformly at random from the interval $[\$0, \$0.5]$ for the air conditioner and water heater. We set the desired load pattern $(x_{a,i,t}^{\text{des}}, t \in \mathcal{T})$ of the air conditioner to a 16-hour period, during which the appliance turns on for an hour and turns off in the next hour in a periodic fashion. We set the desired load pattern of the water heater to a 5-hour period without interruption. To simulate the non-interruptible appliances, we consider

TABLE I
Operating Specifications of Controllable Appliances.

Appliance	$(p_{a,i}^{\text{avg}},\ d_{a,i},\ d_{a,i}^{\min},\ d_{a,i}^{\max})$
Dish washer	(1.5 kW, 2 hr, −, −)
Washing machine	(2.5 kW, 3 hr, −, −)
Stove	(3 kW, 3 hr, −, −)
EV	$(3\text{ kW}, -, (B_i^{\text{d}} - B_i^0)/p_{a,i}^{\text{avg}}, (B_i^{\max} - B_i^0)/p_{a,i}^{\text{avg}})$
Air conditioner	(1.5 kW, −, 2 hr, 8 hr)
Water heater	(2.5 kW, −, 0 hr, 5 hr)

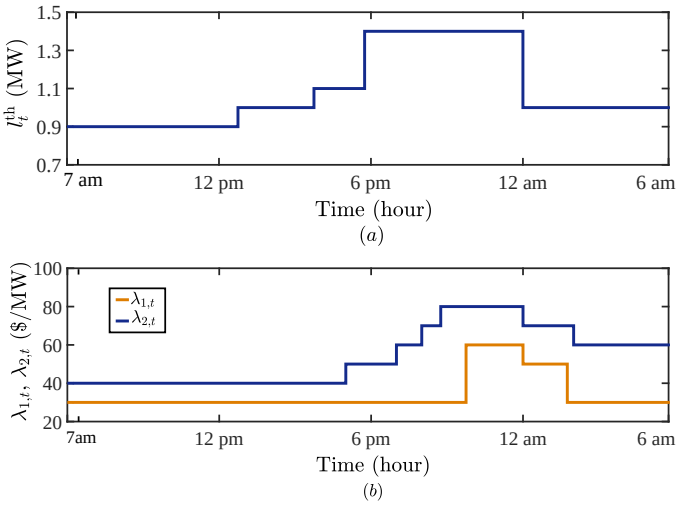


Fig. 2. Price parameters over one day: (a) l_t^{th}; (b) $\lambda_{1,t}$ and $\lambda_{2,t}$.

several scheduling windows selected uniformly between 10 am and 10 pm, with a length that is uniformly chosen at random from set $\{4\,\text{hr}, 5\,\text{hr}, 6\,\text{hr}, 7\,\text{hr}\}$. For the washing machine, we model $(P_{a,i}(\Delta),\ \Delta \geq 1)$ as a truncated normal distribution which is lower bounded by zero, and has a mean value of 288 time slots and a standard deviation of 60 time slots. For other appliances, we use a truncated normal distribution with a mean value of 96 time slots and a standard deviation of 20 time slots. In practical implementations, the probability distribution $(P_{a,i}(\Delta),\ \Delta \geq 1)$ for each appliance a can be approximated by using the historical record on the usage behaviour of each user i.

Unless stated otherwise, the price parameters vary periodically with a period of one day. As discussed in Section II-B, the periodic price parameter vector is a special case for the hidden Markov model in Assumption 1. Figs. 2 (a) and (b) show l_t^{th}, $t \in \mathcal{T}$, and $\lambda_{1,t}$ and $\lambda_{2,t}$, $t \in \mathcal{T}$ over one day, respectively. The actor and critic step sizes in iteration k of the LSL algorithm are set to $\gamma_a^k = m_a/k^{\frac{2}{3}}$ and $\gamma_c^k = m_c/k$, respectively. Since each ECC may use different values for m_a and m_c in practice, we choose m_a and m_c uniformly from $[0.5,\ 2]$ for each household. Unless stated otherwise, the discounted factor β is set to 0.995, i.e., the users are foresighted.

For the benchmark scenario without demand response, the non-interruptible appliances are operated as soon as they become awake. The air conditioner and water heater are operated

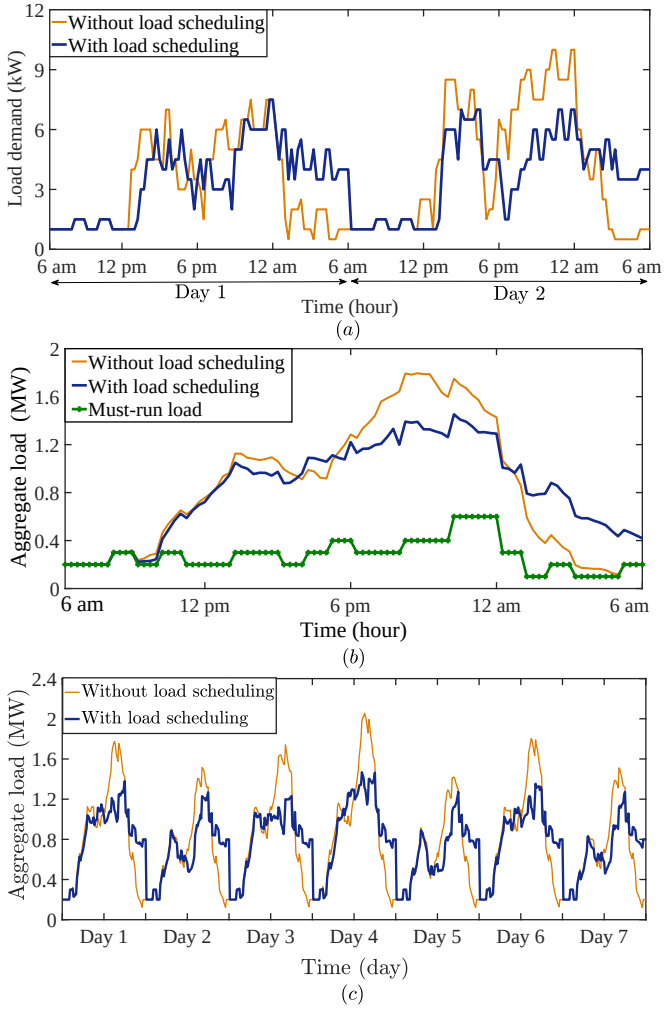


Fig. 3. (a) Load demand for household 1 over two days; (b) aggregate load demand of users over one day; (c) aggregate load demands of all users over seven days with and without load scheduling.

according to their desired load patterns. The EV starts to charge when it is plugged in. We simulate both the benchmark case and LSL algorithm for several scenarios using Matlab in a PC with processor Intel Core i5 3337U CPU 1.80 GHz.

First, we compare the load profiles for household 1 over two days in the benchmark scenario (without load scheduling) and the LSL algorithm (with load scheduling) in Fig. 3 (a). The EV charging demands of household 1 in the first and second days are 6 and 8 hours, respectively. With the LSL algorithm, the ECC of household 1 schedules the operating appliances to reduce the payment. In particular, since the peak load with scheduling in the first day is much lower than that in the second day, the ECC of the foresighted household 1 charges the EV for 8.5 hours in the first day (larger than the demand of 6 hours in the first day), in order to reduce the charging hour to 5.5 hours in the second day. Such a charging schedule reduces the peak load in the second day. Fig. 3 (b) shows the aggregate load demand of all users during one sample day. The peak load is about 1.9 MW around 8 pm without load scheduling. When the households deploy LSL algorithm, the ECCs schedule the controllable appliances to off-peak hours

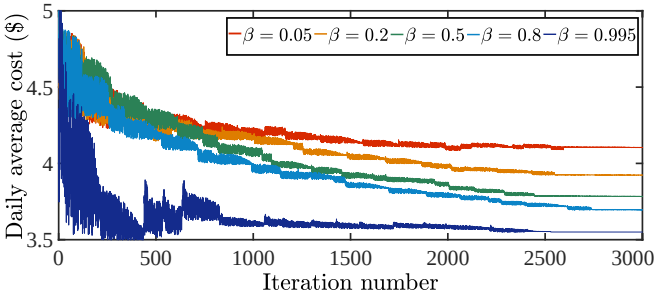


Fig. 4. Daily average cost for myopic and foresighted household 1.

at the MPE. The peak load decreases by 27% to 1.4 MW. Fig. 3 (c) shows the aggregate load profile of all users over one week. The peak load reduction can be observed in all days with the LSL algorithm.

The LSL algorithm benefits the users by reducing their daily average cost. We perform simulations for $\beta = 0.995, 0.8, 0.5, 0.2, 0.05$, which includes the extreme cases of foresighted users ($\beta = 0.995$) and myopic users ($\beta = 0.05$). We present the daily average cost of household 1 for different values of β in Fig. 4. The initial value of \$4.8 per day is the daily average cost without load scheduling. When household 1 is foresighted, its daily average cost decreases by 28% (from \$4.8 per day to \$3.5 per day). When β decreases, the daily average cost increases gradually. For a myopic user, the daily average cost decreases by 11% (from \$4.8 per day to \$4.3 per day). The reason is that the ECC for foresighted users schedules the appliances considering the price in the current and future time slots. Fig. 5 (a) shows the charging profile of the EV for household 1 with a myopic user. Fig. 5 (b) shows the dynamics of electricity price over two days when the users are myopic. The ECC of the myopic user (with $\beta = 0.05$) considers the daily price fluctuations and charges the EV just to fulfill the charging demand (for 6 hours). Fig. 5 (c) shows the charging profile of the EV for household 1 with a foresighted user. Fig. 5 (d) shows the dynamics of electricity price when the users are foresighted. The ECC of the foresighted user (with $\beta = 0.995$), on the other hand, takes advantage of the price fluctuations over multiple days (in particular the low current price) and charges the EV more than the current charging demand (for 8.5 hours) in order to reduce cost in the following day when the price in the charging period is high.

The LSL algorithm helps the utility company reduce the PAR in the aggregate load demand. We compute the expected PAR over a period of 2 months in Fig. 6. We consider two special cases of the hidden Markov model in Assumption 1, i.e., the periodic and random price parameters, respectively, to evaluate the performance of LSL algorithm. With periodic price parameters, the LSL algorithm performs well and reduces the PAR from 2.3 to 2.02 (13% reduction) in 3000 time slots (about a month). For random price parameters, we assume that the utility company chooses l_t^{th}, $t \in \mathcal{T}$ from a truncated normal distribution with a mean value shown in Fig. 2 (a) and a standard deviation of 0.2 MW. The parameters $\lambda_{1,t}$ and $\lambda_{2,t}$, $t \in \mathcal{T}$ are also chosen from a truncated normal distribution with a mean value shown in Fig. 2 (b) and a

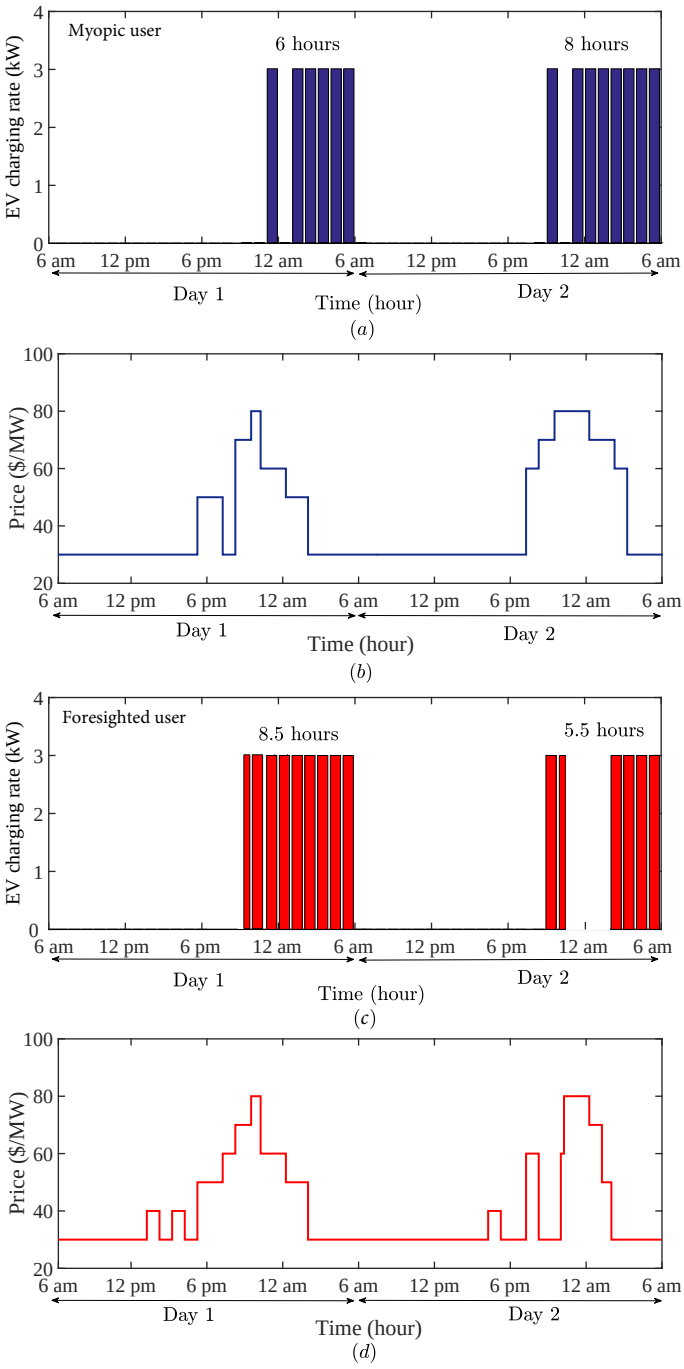


Fig. 5. (a) The EV's charging schedule when household 1 is myopic ($\beta = 0.05$); (b) the electricity price when users are myopic; (c) the EV's charging schedule when household 1 is foresighted ($\beta = 0.995$); (d) The electricity price when users are foresighted.

standard deviation of 5 \$/MW. The random price parameters can model abnormal fluctuations (such as spikes in the price values). In practice, the probability distributions for the price parameters can be estimated from the historical price data. Nevertheless, our LSL algorithm is model-free, hence the ECCs do not need to know the probability distributions of the price parameters. Results shows that the ECCs can still effectively determine their MPE policies through learning, but it takes 6500 time slots (about two months) for the

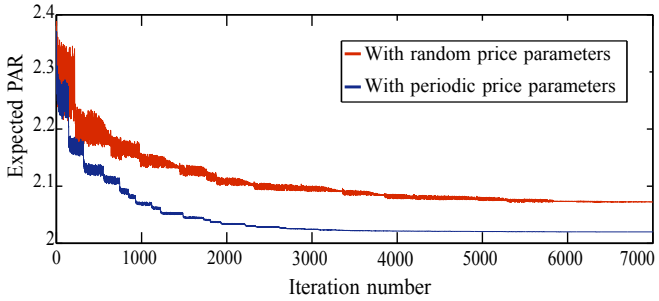


Fig. 6. Expected PAR of the LSL algorithm with periodic and random price parameters.

PAR to converge to 2.07. Thus, LSL algorithm has a robust performance even in a market with random fluctuations in the price parameters.

We show that Algorithm 2 converges to the MPE by using the MPE characterization in Theorem 2. Fig. 7 depicts the absolute values of the approximate objective function $||\widehat{f}_i^{\text{obj}}(V_i^{\boldsymbol{\pi},k}, \pi_i^k)||$ for households 1, 2, and 3. It shows that the objective values converge to zero (we have the same result for other households), which is the global optimal solution of problem (23). Thus from Theorem 2, the LSL algorithm converges to the MPE of Game 2. Though the action and state spaces of each household are large, the speed of convergence is acceptable as a result of using the value function and policy approximations. The jumps in the curves in Fig. 7 correspond to the iterations where the basis functions in (31) and (32) are added to the basis function sets. In our simulation, the running time of the LSL algorithm per iteration per household is only a few seconds. As the households only need to go through one iteration of computation per time slot (e.g., 15 mins), the proposed algorithm is suitable for real-time executions.

We compare the LSL algorithm with a scheduling algorithm based on Q-learning to demonstrate the benefit of the actor-critic method. Q-learning has been used in some existing learning algorithms for demand response (e.g., [7] and [8]). We consider an algorithm based on Q-learning with the same structure as LSL algorithm, with the only difference that the ECC updates the Q-functions [22, Ch. 6]. Fig. 8 shows the daily average cost of household 1 using the LSL algorithm and the Q-learning benchmark. In each iteration of the Q-learning benchmark, the policies are obtained from the updated values of the Q-functions (which is computed based on the Boltzmann exploration as in [7]). The policy update suffers from high fluctuations and slow learning. Our proposed algorithm converges much smoother, with a total convergence time around 25% of that of the Q-learning benchmark.

To study how the observation profile approximation in Algorithm 1 affects the users' policy, we compare the households' policies in two scenarios. In the first scenario, the states are partially observable to the ECCs. They will use Algorithm 1 to approximate the observation profile of all households. In the second scenario, the utility company shares the state of all households with each ECC. Thus, the states become fully observable to the ECCs. The LSL algorithm can be used in both scenarios to determine the MPE policy of the households.

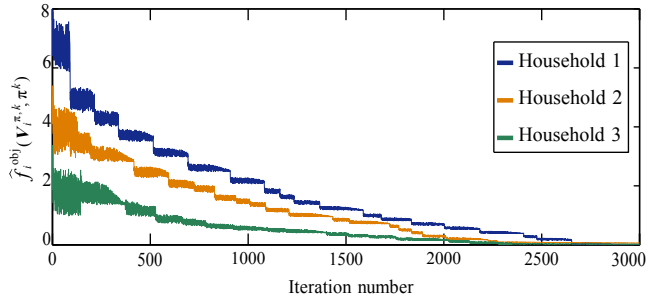


Fig. 7. Objective value $||f_i^{\text{obj}}(V_i^{\boldsymbol{\pi},k}, \pi_i^k)||$ for households 1, 2, and 3.

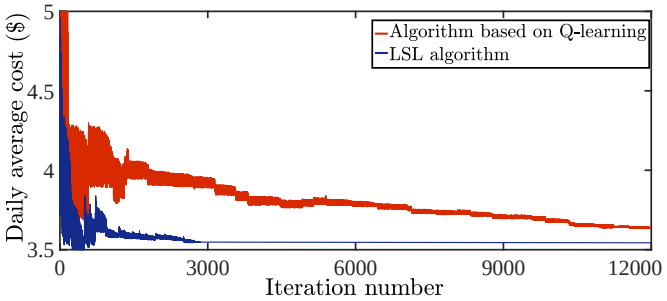


Fig. 8. Daily average cost for household 1 with the algorithm based on Q-learning and our proposed LSL algorithm.

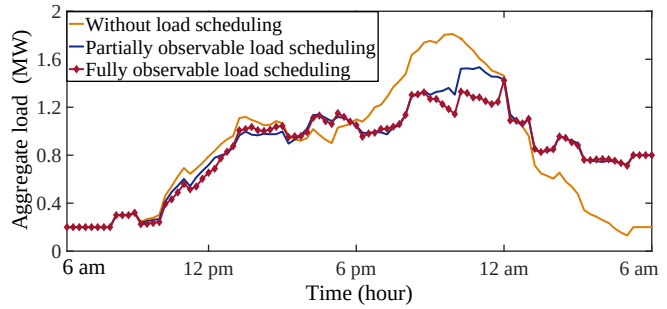


Fig. 9. The aggregate load demand with the partially observable load scheduling and fully observable load scheduling.

Fig. 9 shows the aggregate load demand in both scenarios over one day, with and without load scheduling. When the states are partially observable, the ECCs play a sequence of Bayesian games in Game 2. As each ECC has incomplete information about other households' states, it determines an optimal policy that minimizes the expected cost in all possible states of other households under a given approximate observation profile. When the states are fully observable, the ECCs play a sequence of normal form games. As each ECC knows the actual state of other households, its policy becomes the best response for the actual state of the system. Fig. 9 shows that when the states become fully observable, the peak in the aggregate load demand further decreases when the aggregate load is around the threshold l_t^{th}. This reduces the expected cost of the households, e.g., the daily average cost of household 1 is reduced by 6.3% (from \$3.5 per day to \$3.28 per day).

VI. Conclusion

In this paper, we formulated the scheduling problem of the controllable appliances in the residential households as a partially observable stochastic game, where each household aims at minimizing its discounted average cost in a real-time pricing market. We proposed a distributed and model-free learning algorithm based on the actor-critic method to determine the MPE policy. We used the value function and policy approximation technique to reduce the action and state spaces of the households and improve the learning speed. Simulation results show that the expected PAR in the aggregate load can be reduced by 13% when users deploy the proposed algorithm. Furthermore, the foresighted users can benefit from 28% reduction in their expected discounted cost in long-term, which is 17% lower than the expected cost of the myopic users. For future work, we plan to extend our LSL algorithm to a deregulated market, where multiple households participate in demand response program and can choose to purchase electricity from multiple utility companies.

Appendix

A. The Proof of Equation (1)

Consider appliance $a \in \mathcal{A}_i$ in household i. According to Definition 1, $\delta_{a,i,t}$ for $t \in \mathcal{T}$ is the number of time slots since the most recent time slot that appliance a becomes awake with the most recent new task. In other words, appliance a has not become awake with a new task *again* in time slots $t-\delta_{a,i,t}+1,\ldots,t$ since it became awake in time slot $t-\delta_{a,i,t}+1$. The value of $P_{a,i}(\delta_{a,i,t})$ is the probability that the difference between two sequential wake-up times for appliance a is $\delta_{a,i,t}$. Given the current time slot t, the probability $P_{a,i,t+1}$ that appliance $a \in \mathcal{A}_i$ becomes awake with a new task in the next time slot $t+1 \in \mathcal{T}$ can be obtained from the Bayes' rule as

$$P_{a,i,t+1} = \frac{\text{Prob}\{\text{E}_1 \,|\, \text{E}_2\}\,\text{Prob}\{\text{E}_2\}}{\text{Prob}\{\text{E}_1\}}, \tag{34}$$

where E_1 is the event that appliance a has not become awake with a new task until time slot t, and E_2 is the event that appliance a becomes awake in time slot $t+1$ after $\delta_{a,i,t}$ time slots since it became awake with the most recent task. With probability $\text{Prob}\{\text{E}_1 \,|\, \text{E}_2\} = 1$, appliance a has not become awake with a new task until time slot t conditioned on the event that it becomes awake with a new task in time slot $t+1$. With probability $\text{Prob}\{\text{E}_2\} = P_{a,i}(\delta_{a,i,t})$, appliance a becomes awake in time slot $t+1$ after $\delta_{a,i,t}$ time slots since it became awake with the most recent task. Appliance a has not become awake in time slots $t-\delta_{a,i,t}+1,\ldots,t$ with probability $\text{Prob}\{\text{E}_1\} = 1-\sum_{\Delta=1}^{\delta_{a,i,t}-1} P_{a,i}(\Delta)$. Therefore, $P_{a,i,t+1}$ can be obtained as (1). This completes the proof. ■

B. The Proof of Theorem 1

The MPE policy in Game 2 is the fixed point solution of every household's best response policy. Household i solves the Bellman equations (20) for all approximate observation profiles $\hat{\boldsymbol{o}} \in \mathcal{O}$ when other households' policies are fixed. We construct a Bayesian game from the underlying fully observable game with incomplete information as follows:

Game 3 *Bayesian Game Among Virtual Households*:

Players: The set of virtual households, where each virtual household $(i, \hat{\boldsymbol{o}})$ corresponds to each real household $i \in \mathcal{N}$ and observation profile $\hat{\boldsymbol{o}} \in \mathcal{O}$.

Types: The type of each virtual household $(i, \hat{\boldsymbol{o}})$ is the observation $\boldsymbol{o}_i \in \mathcal{O}_i$ of household i. $P_i(\boldsymbol{o}_i|\hat{\boldsymbol{o}})$ is the probability that virtual household $(i, \hat{\boldsymbol{o}})$ has type $\boldsymbol{o}_i$.

Strategies: The strategy for virtual household $(i, \hat{\boldsymbol{o}})$ is the probability distribution $\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})$ over the actions $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$.

Costs: The cost of each virtual household $(i, \hat{\boldsymbol{o}})$ with strategy $\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})$ is equal to $\mathbb{E}_{\boldsymbol{\pi}_i(\hat{\boldsymbol{o}})}\Big\{Q_i^{\boldsymbol{\pi}_{-i}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\Big\}$, where $Q_i^{\boldsymbol{\pi}_{-i}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ is defined in (18).

We consider the Bayesian Nash equilibrium (BNE) solution concept for the underlying Bayesian game among virtual households. We show that the BNE corresponds to the MPE of Game 2 among households $i \in \mathcal{N}$. In Game 3, each virtual household $(i, \hat{\boldsymbol{o}})$ aims to determine its BNE strategy $\boldsymbol{\pi}_i^{\text{BNE}}(\hat{\boldsymbol{o}})$ to minimize $\mathbb{E}_{\boldsymbol{\pi}_i^{\text{BNE}}(\hat{\boldsymbol{o}})}\Big\{Q_i^{\boldsymbol{\pi}_{-i}^{\text{BNE}}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\Big\}$ when other virtual households' strategies are fixed. Therefore, in the BNE all virtual households solve the Bellman equations in (20). Consequently, the BNE of the Game 3 among virtual households corresponds to the MPE of Game 2 among real households. A BNE always exists for the Bayesian games with a finite number of players and actions [17, Ch. 6]. Thus, an MPE exists for the fully observable game with incomplete information among households. This completes the proof. ■

C. The Proof of Theorem 2

We use an approach similar to [9, Theorem 3.8.2] to show that the joint policy $\boldsymbol{\pi}$ is an MPE if and only if $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i) = 0$ for all households $i \in \mathcal{N}$. Then, we obtain the condition in (24) for the policy in an MPE. Our proof involves two steps.

Step (a) Consider the joint policy $\boldsymbol{\pi}$ and value functions $V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}}), i \in \mathcal{N}$, in the feasible set of problem (23), for which we have $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i) = 0$ for $i \in \mathcal{N}$. We show that the policy $\boldsymbol{\pi}$ is an MPE. According to the constraint set of problem (23), the Bellman errors for the actions in an approximate observation profile $\hat{\boldsymbol{o}}$ are non-negative. Since $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i)$ is the expectation over the Bellman errors, its value is nonnegative for all feasible policies and value functions. If $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i) = 0$ for all $i \in \mathcal{N}$, then the policy $\boldsymbol{\pi}$ and the value functions $V_i^{\boldsymbol{\pi}}(\hat{\boldsymbol{o}})$, $i \in \mathcal{N}$ are the global optimum of problem (23) for all households. Hence, no household has the incentive to unilaterally change its policy, in order to further reduce its objective value $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}}, \boldsymbol{\pi}_i)$. In other words, the policy $\boldsymbol{\pi}$ is an MPE.

Next, we show that for an MPE policy $\boldsymbol{\pi}^{\text{MPE}}$, we can determine a value function $V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}})$ such that $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = 0$. From (22), $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = 0$ is equivalent to

$$\sum_{\hat{\boldsymbol{o}} \in \mathcal{O}} \mathbb{E}_{\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})}\Big\{B_i\left(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right)\Big\} = 0, \quad \forall\, i \in \mathcal{N}. \tag{35}$$

According to the constraint set of problem (23), the Bellman errors for the actions in an observation profile $\hat{\boldsymbol{o}}$ are nonnegative in the MPE. Thus, each term of the summation in (35) should be zero. That is

$$\mathbb{E}_{\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})}\Big\{B_i\left(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right)\Big\} = 0, \ \ \forall\, \hat{\boldsymbol{o}} \in O, \forall\, i \in \mathcal{N}, \tag{36}$$

which is equivalent to

$$\mathbb{E}_{\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})}\Big\{Q_i^{\boldsymbol{\pi}_{-i}^{\text{MPE}}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})) - V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}})\Big\} = 0, \forall\, \hat{\boldsymbol{o}} \in O, \forall\, i \in \mathcal{N}. \tag{37}$$

$\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})$ is a randomized policy. Hence, we have $\mathbb{E}_{\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})}\big\{V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}})\big\} = V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}})$. Hence, for all approximate observation profile $\hat{\boldsymbol{o}} \in O$, (37) can be rewritten as

$$V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}}) = \mathbb{E}_{\boldsymbol{\pi}_i^{\text{MPE}}(\hat{\boldsymbol{o}})}\Big\{Q_i^{\boldsymbol{\pi}_{-i}^{\text{MPE}}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\Big\}, \quad \forall\, i \in \mathcal{N}. \tag{38}$$

For household i, we define the average cost in approximate observation profile $\hat{\boldsymbol{o}}$ as $\bar{c}_i(\hat{\boldsymbol{o}}) = \mathbb{E}_{\boldsymbol{\pi}^{\text{MPE}}(\hat{\boldsymbol{o}})}\{c_i(\hat{\boldsymbol{o}}, \boldsymbol{x}(\hat{\boldsymbol{o}}))\}$. We define the average transition probability from observation $\hat{\boldsymbol{o}}$ to $\hat{\boldsymbol{o}}'$ as $\bar{P}(\hat{\boldsymbol{o}}' \,|\, \hat{\boldsymbol{o}}) = \mathbb{E}_{\boldsymbol{\pi}^{\text{MPE}}(\hat{\boldsymbol{o}})}\{P(\hat{\boldsymbol{o}}' \,|\, \hat{\boldsymbol{o}}, \boldsymbol{x}(\hat{\boldsymbol{o}}))\}$. We define vectors $\bar{\boldsymbol{c}}_i = (\bar{c}_i(\hat{\boldsymbol{o}}),\, \hat{\boldsymbol{o}} \in O)$ and $\boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}} = \big(V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}}),\, \hat{\boldsymbol{o}} \in O\big)$, and define the transition matrix $\bar{\boldsymbol{P}} = \big[\bar{P}(\hat{\boldsymbol{o}}' \,|\, \hat{\boldsymbol{o}}),\, \hat{\boldsymbol{o}}, \hat{\boldsymbol{o}}' \in O\big]$. By substituting (18) into (38), we have

$$\boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}} = (1-\beta)\bar{\boldsymbol{c}}_i + \beta \bar{\boldsymbol{P}} \boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}}. \tag{39}$$

By rearranging the terms in (39), we obtain

$$\left(\boldsymbol{I} - \beta \bar{\boldsymbol{P}}\right) \boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}} = (1-\beta)\bar{\boldsymbol{c}}_i, \tag{40}$$

where $\boldsymbol{I}$ is the identity matrix. Matrix $\bar{\boldsymbol{P}}$ is a stochastic matrix (i.e., each of its entries is a nonnegative real number representing a probability), and thus its eigenvalues are less than or equal to one. Besides, the discount factor β is less than one. Hence, the eigenvalues of matrix $\boldsymbol{I} - \beta \bar{\boldsymbol{P}}$ are positive, and thereby it is invertible (or nonsingular). From (40), we can obtain $\boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}}$ as

$$\boldsymbol{V}_i^{\boldsymbol{\pi}^{\text{MPE}}} = (1-\beta)\left(\boldsymbol{I} - \beta \bar{\boldsymbol{P}}\right)^{-1} \bar{\boldsymbol{c}}_i. \tag{41}$$

Therefore, for the MPE policy $\boldsymbol{\pi}^{\text{MPE}}$, we obtain the value function $V_i^{\boldsymbol{\pi}^{\text{MPE}}}(\hat{\boldsymbol{o}})$ in (41) such that $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = 0$ for all households $i \in \mathcal{N}$.

Step (b) We obtain the condition in (24) for the policy in an MPE. For each household $i \in \mathcal{N}$, the objective function $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}})$ in (22) can be expressed as

$$f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = \sum_{\hat{\boldsymbol{o}} \in O} \sum_{\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})} \pi_i^{\text{MPE}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\, B_i\left(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right). \tag{42}$$

The Bellman error $B_i(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))$ is nonnegative. Hence, from (42), $f_i^{\text{obj}}(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \boldsymbol{\pi}_i^{\text{MPE}}) = 0$ is equivalent to $\pi_i^{\text{MPE}}(\hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}))\, B_i\left(V_i^{\boldsymbol{\pi}^{\text{MPE}}}, \hat{\boldsymbol{o}}, \boldsymbol{x}_i(\hat{\boldsymbol{o}})\right) = 0$ for all households $i \in \mathcal{N}$ with action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}) \in \hat{\mathcal{X}}_i(\hat{\boldsymbol{o}})$ in observation profile $\hat{\boldsymbol{o}}$. This completes the proof. ■

D. Bellman Error Approximation

The basis function in (32) is equal to the expectation over the Bellman errors for all feasible actions

$\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. ECC i knows the observation $\boldsymbol{o}_i^{k-1}$, the approximate observation profile $\hat{\boldsymbol{o}}^{k-1}$, and the cost $c_i\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{x}_{-i}^{k-1}(\hat{\boldsymbol{o}}^{k-1})\right)$ for the chosen action $\boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1})$ in iteration $k-1$, as well as the current observation $\boldsymbol{o}_i^k$ and the approximate observation profile $\hat{\boldsymbol{o}}^k$. ECC i needs to use these available information to approximate the Bellman error for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. We use the TD error as an estimation for the Bellman error [14, Lemma 3]. We have

$$\begin{aligned} & B_i^{k-1}\left(V_i^{\boldsymbol{\pi},k-1}, \hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\right) \\ & \approx (1-\beta)\, c_i\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{x}_{-i}^{k-1}(\hat{\boldsymbol{o}}^{k-1})\right) \\ & \quad + \beta\, V_i^{\boldsymbol{\pi},k-1}\left(\hat{\boldsymbol{o}}^k\left(\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\right), \boldsymbol{\theta}_i^{k-1}\right) \\ & \quad - V_i^{\boldsymbol{\pi},k-1}\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{\theta}_i^{k-1}\right), \end{aligned} \tag{43}$$

where $\hat{\boldsymbol{o}}_i^k\left(\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\right)$ is the approximate observation profile in the current time slot k if household i chooses action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})$ in the previous time slot $k-1$. ECC i determines $\hat{\boldsymbol{o}}_i^k\left(\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\right)$ in the following two steps:

Step (a) ECC i knows observations $\boldsymbol{o}_i^{k-1}$ and $\boldsymbol{o}_i^k$. Thus it can determine the set of appliances that become awake with a new task in the current time slot k. ECC i can also determine the state of other operating appliances for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})$. Therefore, ECC i can determine the state of its own household for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})$.

Step (b) The states of other households are fixed. Furthermore, ECC i knows the approximate observation profile $\hat{\boldsymbol{o}}^k$ for the chosen action $\boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1})$. Using the result of Step (a), ECC i can compute the average aggregate load demands for the feasible actions of all households for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$, and thus it can determine the approximate observation profile $\hat{o}^k(\boldsymbol{x}_i(\hat{\boldsymbol{o}}^k))$ for all households for action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})$.

In addition to computing the approximate observation profile $\hat{\boldsymbol{o}}_i^k\left(\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})\right)$, ECC i needs to compute the cost $c_i\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{x}_{-i}^{k-1}(\hat{\boldsymbol{o}}^{k-1})\right)$ for feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. ECC i knows the payment to the utility company for the chosen action $\boldsymbol{x}_i^{k-1}(\hat{\boldsymbol{o}}^{k-1})$. Since the load demand of one household is much smaller than the aggregate load demand of all households, we can assume that the price value is unchanged when household i unilaterally changes its load demand. Thus, ECC i can estimate its payment for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. ECC i can also determine the discomfort cost for action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}) \in \mathcal{X}_i(\boldsymbol{o}_i^{k-1})$. Therefore, it can compute the cost $c_i\left(\hat{\boldsymbol{o}}^{k-1}, \boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1}), \boldsymbol{x}_{-i}^{k-1}(\hat{\boldsymbol{o}}^{k-1})\right)$ for an arbitrary feasible action $\boldsymbol{x}_i(\hat{\boldsymbol{o}}^{k-1})$. Finally, ECC i is able to compute the approximate Bellman error in (43).

REFERENCES

[1] Office of Electricity Delivery & Energy Reliability, "Customer participation in the smart grid: Lessons learned," U.S. Department of Energy, Tech. Rep., Sept. 2014.

[2] The Brattle Group, Freeman, Sullivan & Co., and Global Energy Partners, LLC, "A national assessment of demand response potential," Federal Energy Regulatory Commission, Tech. Rep., Jun. 2009.

[3] P. Samadi, A. Mohsenian-Rad, V.W.S. Wong, and R. Schober, "Real-time pricing for demand response based on stochastic approximation," *IEEE Trans. on Smart Grid*, vol. 5, no. 2, pp. 789–798, Mar. 2014.

[4] Z. Chen, L. Wu, and Y. Fu, "Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization," *IEEE Trans. on Smart Grid*, vol. 3, no. 4, pp. 1822–1831, Dec. 2012.

[5] C. Eksin, H. Delic, and A. Ribeiro, "Demand response management in smart grids with heterogeneous consumer preferences," *IEEE Trans. on Smart Grid*, vol. 6, no. 6, pp. 3082–3094, Nov. 2015.

[6] N. Forouzandehmehr, M. Esmalifalak, A. Mohsenian-Rad, and Z. Han, "Autonomous demand response using stochastic differential games," *IEEE Trans. on Smart Grid*, vol. 6, no. 1, pp. 291–300, Jan. 2015.

[7] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 6, no. 5, pp. 2312–2324, Sept. 2015.

[8] B. Kim, Y. Zhang, M. van der Schaar, and J. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Trans. on Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sept. 2016.

[9] Y. Liang, L. He, X. Cao, and Z. J. Shen, "Stochastic control for smart grid users with flexible demand," *IEEE Trans. on Smart Grid*, vol. 4, no. 4, pp. 2296–2308, Dec. 2013.

[10] F. Ruelens, B. J. Claessens, S. Vandael, B. D. Schutter, R. Babuska, and R. Belmans, "Residential demand response of thermostatically controlled loads using batch reinforcement learning," accepted for publication in *IEEE Trans. on Smart Grid*, 2016.

[11] Y. Xiao and M. van der Schaar, "Distributed demand side management among foresighted decision makers in power networks," in *Proc. of IEEE Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2013.

[12] J. Yao and P. Venkitasubramaniam, "Optimal end user energy storage sharing in demand response," in *Proc. of IEEE SmartGridComm*, Miami, FL, Nov. 2015.

[13] L. Jia, Q. Zhao, and L. Tong, "Retail pricing for stochastic demand with unknown parameters: An online machine learning approach," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Oct. 2013.

[14] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. NY: Springer, 1997.

[15] E. Kalai and E. Lehrer, "Rational learning leads to Nash equilibrium," *Econometrica*, vol. 39, no. 10, pp. 1019–1045, Jul. 1993.

[16] A. Sandroni, "Does rational learning lead to Nash equilibrium in finitely repeated games?" *Journal of Economic Theory*, vol. 78, no. 1, pp. 195 – 218, 1998.

[17] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *Proc. of Int'l Conf. on Artificial Intelligence*, Seattle, WA, Aug. 2001.

[18] L. M. Dermed and C. L. Isbell, "Solving stochastic games," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1186–1194. [Online]. Available: http://papers.nips.cc/paper/3825-solving-stochastic-games.pdf

[19] L. Li and J. Shamma, "LP formulation of asymmetric zero-sum stochastic games," in *Proc. of IEEE Annual Conf. on Decision and Control*, Los Angeles, CA, Dec. 2014.

[20] R. N. Borkovsky, U. Doraszelski, and Y. Kryukov, "A user's guide to solving dynamic stochastic games using the homotopy method," *Operations Research*, vol. 58, no. 4-part-2, pp. 1116–1132, Jul. 2010.

[21] M. Neely, "A Lyapunov optimization approach to repeated stochastic games," in *Proc. of Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, Oct. 2013.

[22] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[23] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Approximate solutions for partially observable stochastic games with common payoffs," in *Proc. of Int'l Conf. on Autonomous Agents and Multiagent Systems*, New York, NY, Jul. 2004.

[24] F. Oliehoek, S. Whiteson, and M. Spaan, "Approximate solutions for factored Dec-POMDPs with many agents," in *Proc. of Int'l Conf. on Autonomous Agents and Multiagent Systems*, Saint Paul, MN, May 2013.

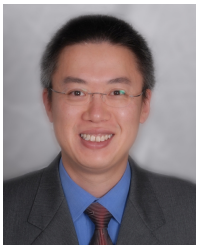
[25] L. MacDermed, C. Isbell, and L. Weiss, "Markov games of incomplete information for multi-agent reinforcement learning," in *Proc. of Int'l Conf. on Artificial Intelligence*, San Fransico, CA, Aug. 2011.

[26] S. Bahrami and V.W.S. Wong, "An autonomous demand response program in smart grid with foresighted users," in *Proc. of IEEE SmartGridComm*, Miami, FL, Nov. 2015.

[27] I. H. Witten, "An adaptive optimal controller for discrete-time Markov environments," *Information and Control*, vol. 34, no. 4, pp. 286–295, Aug. 1977.

[28] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, no. 5, pp. 834–846, Sept. 1983.
[29] V. Konda and J. Tsitsiklis, "On actor-critic algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp. 1143–1166, Aug. 2003.
[30] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, Nov. 2009.
[31] A. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 120–133, Sept. 2010.
[32] D. W. Bunn, *Modelling Prices in Competitive Electricity Markets*. Tornoto, Canada: Wiley Finance, 2004.
[33] R. S. Mamon and R. J. Elliott, *Hidden Markov Models in Finance*. NY: Springer, 2014.
[34] P. Yang, G. Tang, and A. Nehorai, "A game-theoretic approach for optimal time-of-use electricity pricing," *IEEE Trans. on Power Systems*, vol. 28, no. 2, pp. 884–892, Aug. 2013.
[35] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.
[36] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. FL: CRC Press, 2009.
[37] R. Parr, C. Painter-Wakefield, L. Li, and M. L. Littman, "Analyzing feature generation for value-function approximation," in *Proc. of Int'l Conf. on Machine Learning*, New York, NY, Jun. 2007.
[38] Toronto Hydro. [Online]. Available: http://www.torontohydro.com/sites/electricsystem/residential/yourbilloverview/Pages/ApplianceChart.aspx

Shahab Bahrami (S'12) received the B.Sc. and M.A.Sc. degrees both from Sharif University of Technology, Tehran, Iran, in 2010 and 2012, respectively. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, The University of British Columbia (UBC), Vancouver, BC, Canada. His research interests include optimal power flow analysis, game theory, and demand side management, with applications in smart grid.



Vincent W.S. Wong (S'94, M'00, SM'07, F'16) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microsemi). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, and the Internet. Dr. Wong is an Editor of the *IEEE Transactions on Communications*. He was a Guest Editor of *IEEE Journal on Selected Areas in Communications* and *IEEE Wireless Communications*. He has served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. He has served as a Technical Program Co-chair of *IEEE SmartGridComm'14*, as well as a Symposium Co-chair of *IEEE SmartGridComm'13* and *IEEE Globecom'13*. Dr. Wong is the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications and the IEEE Vancouver Joint Communications Chapter. He received the 2014 UBC Killam Faculty Research Fellowship.

Jianwei Huang (S'01-M'06-SM'11-F'16) is an IEEE Fellow, a Distinguished Lecturer of IEEE Communications Society, and a Thomson Reuters Highly Cited Researcher in Computer Science. He is an Associate Professor and Director of the Network Communications and Economics Lab (ncel.ie.cuhk.edu.hk), in the Department of Information Engineering at the Chinese University of Hong Kong. He received the Ph.D. degree from Northwestern University in 2005, and worked as a Postdoc Research Associate at Princeton University during 2005-2007. He is the co-recipient of 8 Best Paper Awards, including IEEE Marconi Prize Paper Award in Wireless Communications in 2011. He has co-authored six books, including the textbook on "Wireless Network Pricing". He received the CUHK Young Researcher Award in 2014 and IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2009. He has served as an Associate Editor of IEEE/ACM Transactions on Networking, IEEE Transactions on Wireless Communications, and IEEE Journal on Selected Areas in Communications - Cognitive Radio Series, and IEEE Transactions on Cognitive Communications and Networking. He has served as the Chair of IEEE ComSoc Cognitive Network Technical Committee and Multimedia Communications Technical Committee.