Structure-aware Stochastic Load Management in Smart Grids

Yu Zhang
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095, USA
yuzhang@ucla.edu

Mihaela van der Schaar
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095, USA
mihaela@ee.ucla.edu

Abstract— **Load management based on dynamic pricing has been advocated as a key approach for demand-side management in smart grids. By appropriately pricing energy, economic incentives are given to consumers to shift their usage away from peak hours, thereby limiting the amount of energy that needs to be produced. However, traditional pricing-based load management methods usually rely on the assumption that the statistics of the system dynamics (e.g. the time-varying electricity price, the arrival distribution of consumers' load demands) are known a priori, which is not true in practice. In this paper, we propose a novel price-dependent load scheduling algorithm which, unlike previous works, can operate optimally in systems where such statistical knowledge is unknown. We consider a power grid system where each consumer is equipped with an energy storage device that has the capability of storing electrical energy during peak hours. Specifically, we allow each consumer to proactively determine the amount of energy to purchase from the utility companies (or energy producers) while taking into consideration that its load demand and the electricity price dynamically vary over time in an a priori unknown manner. We first assume that all the dynamics are known and formulate the real-time load scheduling as a Markov decision process and systematically unravel the structural properties exhibited by the resulting optimal load scheduling policy. By utilizing these structural properties, we then prove that our proposed load scheduling algorithm can learn the system dynamics in an online manner and converge to the optimal solution. A distinctive feature of our algorithm is that it actively exploits partial information about the system dynamics so that less information needs to be learned than when using conventional reinforcement learning methods, which significantly improves the adaptation speed and the runtime performance. Our simulation results demonstrate that the proposed load scheduling algorithm achieves efficiency by more than 30% compared to existing state-of-the-art online learning algorithms.**

I. INTRODUCTION

With the rapid progress of information and communication technologies, such as advanced metering, bi-directional communication, distributed power generation and storage, etc., demand-side management (DSM) is prevailing in the smart power grid, which provides effective balancing over the dynamic power supply and load demand in order to ensure efficient use of electric energy [1]. By controlling the consumers' appliances and move some of the non-urgent loads from peak hours to off-peak hours, DSM can effectively alleviate high demand loads of electric energy at peak hours, thereby improving the stability of the power system and lowering production costs in the long run [2].

There exists a large body of literature on DSM, see e.g. [1]-[12]. Depending on who performs the management, existing DSM methods can be generally classified into two categories. The first category of DSM largely relies on the direct load control (DLC) [3][4]. In this case, the utility companies install switches or thermostats on top of the existing metering infrastructure, which allow them to (directly) modify the operations of appliances during peak hours. For instance, [3] studies optimal centralized energy reallocation in smart grids; [4] investigates the coordination of charging plug-in hybrid electric vehicles with other electric appliances. Although utility-based DLC have been effective in smoothing peak demands, they incur frequent interruptions to the normal use of the household appliances, because the control of DLC is based on the observation of the real-time load without considering the actual demand from the consumers. For instance, when warranted by capacity shortage during the summer, a consumer's central air conditioning system will be turned down or cycled by the utility company, while the exact days and the length of the cycling period will not be known by the consumer in advance [8]. More importantly, by providing centralized control over the electricity load, DLC methods usually neglect the heterogeneity embedded in the consumers' demands and shield individual consumers from making price-aware decisions in order to effectively (and more flexibly) perform individual load scheduling based on their personal demands. This further reduces the efficiency of the smart grid.

Due to the abovementioned problems of DLC, the second category of DSM, which is based on dynamic pricing, has become more prominent in recent years [1][2][5]-[12]. The basic principle of dynamic pricing is to adaptively adjust the retail price of electric energy according to the real-time variation of the production capacity of energy producers and the load demands from the consumers. Although the dynamic pricing does not directly control the load scheduling on individual consumers, appropriate pricing can provide effective economic incentives to consumers and thus shift electricity usage away from peak hours, which in turn helps the utility companies to procure electric energy more efficiently [2].

The dynamic pricing literature can be sub-divided into two categories. The first category takes the utility companies perspective and designs effective pricing strategy in order to maximize the social welfare, i.e. the sum benefit of all consumers, in the smart grid, or the revenue of the utility company from electric energy sale [6]-[8]. The second category focuses on individual or groups of consumers, and mainly aims to design effective price-based load/demand scheduling which maximizes the benefit of the individual (or group of) consumers, given the exogenously determined pricing strategy from the utility company, e.g. [9][10].

In this paper, we specifically focus on the design of price-based load scheduling algorithms from the consumer's perspective, while keeping the design of pricing strategies fixed. Most existing price-based load scheduling algorithms have focused on myopically maximizing the immediate benefit

978-14799-3360-0/14/$31.00 ©2014 IEEE

	[3][4]	[6]-[8]	[9][10]	[1][2][11][12]	Our work
DSM approach	DLC	Dynamic pricing	Dynamic pricing	Dynamic pricing	Dynamic pricing
Optimizing entity	Utility company	Utility company	Individual consumer	Individual consumer	Individual consumer
Load scheduling approach	Centralized control	N/A	Deferring non-urgent load	Deferring non-urgent load	Electric energy storage
Optimization criterion	Myopic/Foresighted	Myopic/Foresighted	Myopic	Foresighted	Foresighted
System dynamics	Known	Known	Known	Known	Unknown
Online learning	No	No	No	No	Yes

Table 1 Comparison of the existing literature and our work

of consumers, based on the current electricity price and load demand, e.g. [9][10]. However, in the smart grid, the load scheduling decisions are strongly correlated across time. That is, the current load scheduling decision will not only affect a consumer's immediate benefit, but also its load demand and benefit in the future [2]. Hence, the myopic optimization of the consumer's benefit cannot perform well in the long run.

There are only a few works which design the load scheduling algorithms by considering the foresighted maximization of the consumer's long-run benefit [1][2][11][12]. Most of them use electric price predictions and assume that the statistical knowledge of the underlying system dynamics (e.g. the temporal variation of the electricity prices, the arrival distribution of the consumer's load demand) is known. However, practical smart grid systems face many unknowns, such as the weather, the heterogeneous consumer reactions to real-time prices, the intermittency of renewable energy sources (e.g. small wind farms, household with solar panels, etc.), whose statistical knowledge cannot be reliably obtained a priori. Therefore, the efficacy of the methods proposed in these works, which rely on specific models of the system dynamics, result in poor performance in practice.

In this paper, we propose a price-based load scheduling algorithm which can operate optimally in time-varying unknown environment. In particular, we consider a power grid where each consumer is equipped with an energy storage device that has the capability of storing electrical energy. The consumers purchase electric energy from an electricity market where the electricity price varies over time in an unknown manner. Hence, each individual consumer performs load scheduling by proactively determining how much electric energy to purchase at each moment of time, given its real-time load demand and the electricity price. By rigorously formulating the consumer's decision problem as a Markov Decision Process (MDP), we then propose an efficient online learning algorithm that enables each consumer to learn the optimal scheduling strategy that maximizes its personal benefit in the long run.

The differences between our work and the existing literature on DSM are exhibited in Table 1. The main contributions of our work are summarized in the following aspects:

(1) **Low-complexity online learning**. We assume that both the electricity price and the arrival of consumers' demands vary dynamically over time. Meanwhile, the statistical knowledge of these dynamics is not known a priori. In order to cope with such unknown time-varying system dynamics, we propose, in our load scheduling algorithm, a decomposition of the (offline) value iteration and (online) reinforcement learning based on factoring the system dynamics into a priori known and a priori unknown components. This is achieved by generalizing the concept of a post-decision state [19][20], which is an intermediate state that occurs *after* the known dynamics take place but *before* the unknown dynamics take place. A key advantage of the proposed PDS learning method is that it exploits partial information about the smart grid system and the structure of the load scheduling problem and thus, less information needs to be learned than when using conventional reinforcement learning algorithms such as Q-learning, actor-critic etc. [18]. Importantly, under certain conditions, it obviates the need for action exploration, which significantly improves the adaptation speed and the runtime performance as compared to conventional reinforcement learning algorithms which loose significant performance during the (very long) exploration state.

(2) **Batch update.** We also take advantage of the fact that the unknown environment dynamics are independent of certain components of the system's state. We exploit this property to perform a batch update on multiple PDSs in each time slot. Importantly, our numerical results illustrate that incorporating batch update into the PDS learning can significantly reduce the convergence time to the optimal policy compared to the traditional "one state at a time" update adopted in reinforcement learning.

(3) **Load scheduling with electric energy storage.** Most existing load scheduling methods mainly rely on prioritizing the consumer's task and postponing non-urgent or deferrable tasks, in order to achieve the balance between supply and demand [1]. Different from this, the load scheduling algorithm proposed in this paper makes use of electric energy storage devices, such as uninterrupted power supply (UPS), rechargeable batteries, and plug-in hybrid electric vehicles, which become prevalent in the current smart grids [13]. Following our algorithm these devices are charged at off-peak hours and the stored energy can be used to satisfy increased demand at peak hours. By optimally designing the energy storage policy, our load scheduling algorithm does not need to defer any load demand, which often causes dissatisfaction from individual consumers.

The remainder of this paper is organized as follows. In Section II, a rigorous MDP framework is proposed to formulate the load scheduling problem in the smart grid. In Section III, we describe a novel PDS online learning algorithm that optimally solves the load scheduling problem and study the structure of the optimal load scheduling policy. After presenting the simulation results in Section IV, we conclude in Section V.

II. SYSTEM MODEL

A. System Setting

This section describes the smart grid system assumed in this paper. We consider an infinite-horizon discrete time model,

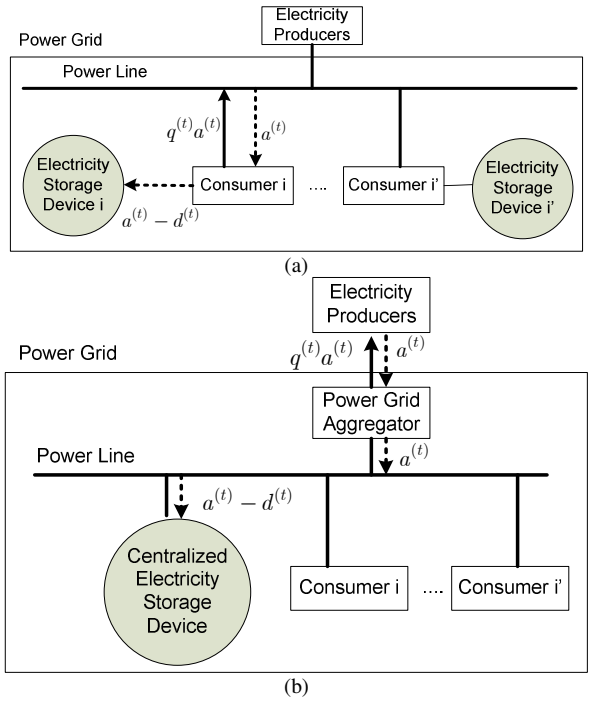

Figure 1 Electricity wholesale market (a) without aggregators; (b) with aggregators

where time slots are indexed by $t = 0,1,\ldots$. Here each slot represents the time interval in which the entities in the system make one operation. For instance in [1], the length of each time slot is assumed to be one hour, in which the sale price of electricity changes once. Similar to [10][12], we consider an electricity market where distributed power grids purchase electric energy from distributed energy producers. A market consists of three different types of agents: market operator, producer, and consumer:

(1) The *producers* represent the distributed entities who generate and sell energy in this wholesale market to the power grids [10][14]. Examples of such producers include small wind farms, households with solar panels, etc.

(2) The *consumers* are the end users residing in the power grids. Each consumer owns a number of residential appliances, e.g. electrical vehicles, air conditions, dishwashers, etc. Each consumer purchases electric energy from the market for its own consumption.

(3) The *market operator* is a monopoly who regulates the market, e.g. the owner of the market. It manages the physical infrastructure and the electricity trading in the market and determines the trading price of the electricity.

We specifically focus in this paper on designing optimal policies for strategic consumers, while assuming that the other entities in the market (i.e. the producers and the market operator) are obedient and follow given policies [1]. Here we assume that the load scheduling policies of different consumers do not interfere with the decisions of each other and hence, it is sufficient to focus on the analysis of one representative consumer.

In each time slot t, the interaction of the consumer with other entities in the market can be summarized as follows:

(1) The market operator publishes the unit electricity price in the current time slot. The unit price is denoted as $q^{(t)} \in \mathcal{Q}$, where $\mathcal{Q}$ is a finite set of possible electricity prices. Similar to [9], we assume that the evolution of the electricity price follows a stationary finite-state Markov chain and the transition probability is determined by $p_q(q^{(t+1)} \mid q^{(t)})$, which is exogenously determined [2].

(2) The consumer observes its load demand, which is denoted as $d^{(t)}$, in the current time slot. We assume that $d^{(t)} \in [0, D]$, where D is a constant.

(3) The consumer purchases an amount $a^{(t)} \in \mathcal{A}$ of electric energy from the electricity producers to fulfill its demand, where $\mathcal{A}$ represents the finite set of possible purchasing amounts.

A significant problem embedded in the electricity market is that the production capacity of the distributed electricity producers varies drastically over time and is often highly unpredictable compared to the large power plants because they rely on intermittent resources like wind and sunshine [9], which in turn introduce significant fluctuations on the unit electricity price $q^{(t)}$. Therefore, the stability of the power grids is critically dependent on having balanced electricity supply and demand at any given time. In order to achieve such balance between the supply and demand, the consumer schedules its load demand through the help of electricity storage devices (e.g. batteries, plug-in hybrid electric vehicle, etc.). With the help of electricity storage devices, the basic principle of the consumer's load scheduling can be described as follows:

- In a time slot t when the production capacity of producers is high and thus the unit electricity price $q^{(t)}$ is low, the consumer purchases more electricity than what is demanded by its appliances, i.e. $a^{(t)} > d^{(t)}$, and stores the *surplus* $a^{(t)} - d^{(t)}$ in the storage device.
- In a time slot t when the production capacity is low and thus the unit electricity price $q^{(t)}$ is high, the consumer purchases less electricity than what is demanded by its appliances, i.e. $a^{(t)} < d^{(t)}$, and covers the deficit $d^{(t)} - a^{(t)}$ using the stored electricity, which is denoted as $b^{(t)} \in \mathcal{B}$, where $\mathcal{B}$ is a countable set.

A schematic representation of the considered electricity market is illustrated in Figure 1(a). By strategically determining its purchased amount $a^{(t)}$ in each time slot, the consumer can flexibly utilize its stored electricity to effectively balance the electricity supply and demand across time slots while minimizing the negative effect introduced by the

[1] It should be noted that the stochastic control and online learning solutions proposed in this paper can be easily extended to the design and analysis of the strategic operations of entities other than the consumers. We relegate such extension as future works.

[2] It should be noted that the Markovian price model we assumed here is only for analytical tractability. As what is shown in Section IV, our proposed load scheduling algorithm also performs well when the price variation is not Markovian.

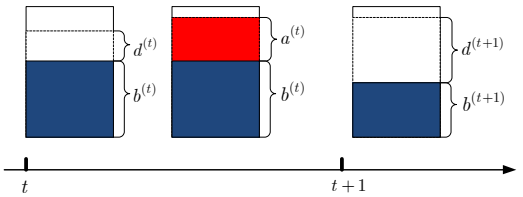

Figure 2 Temporal evolution of the electricity storage

production capacity fluctuation. In this way, its benefit from electricity consumption can be maximized.

B. An Alternative Setting – Load Scheduling by Aggregators

We would like to note that our proposed load scheduling with energy storage devices does not necessarily need to be performed by individual consumers. In certain scenarios, consumers can be grouped together and perform collaborative load scheduling with the help of some aggregators and centralized energy storage devices [13]. For instance, the load scheduling of the centralized air-conditioning in a building is usually not performed individually by users in their separate apartments or rooms but controlled by some centralized building facility manager.

A schematic representation of the electricity market with aggregators is illustrated in Figure 1(b). The job of the aggregator in each time slot is to aggregate the load demands from individual consumers, purchase the electric energy from the market, distribute the demanded energy and store the surplus into the associated centralized energy storage devices. Therefore, $d^{(t)}$ and $a^{(t)}$ in this figure represent the sum demand and purchased electricity of all consumers in a group, instead of the individual demand and purchased electricity.

It is important to note that our analysis throughout the rest of this paper, though being conducted from an individual consumer's perspective, can be applied, without any change, into the design of the optimal load scheduling policy for the aggregator. With aggregators, individual consumers can share the energy storage device and thus reduce the associated cost for energy storage. However, such collaborative load scheduling also introduces negative effects, since the benefit received by each individual consumer will be influenced by the dynamics (e.g. the load demands) at other consumers. In Section IV, we will demonstrate this negative effect using numerical examples.

C. Stochastic Control Formulation

In this section, we formulate the strategic load scheduling of the consumer as a stochastic control problem. The action in each time slot t is its purchased amount $a^{(t)}$. The state is defined as a tuple $s^{(t)} \triangleq (q^{(t)}, b^{(t)}, e^{(t)})$. Here the variable $e^{(t)}$ represents all environment dynamics (e.g. time, weather) that the consumer experiences in time slot t, other than the electricity price $q^{(t)}$, its own demand $d^{(t)}$, and its current stored amount of electricity $b^{(t)}$. Valuable information is embedded in the environment dynamics, which can enable the consumer to make more effective purchasing decisions. For instance, it is frequently the case in an energy market that the sale price may be high during the peak hours (e.g. 6pm-12pm) and low during off-peak hours (e.g. 12pm-6pm) [10][15]. Hence, this information is helpful to consumers aiming to predict how their own demand and the electricity price will change in the near future and, based on this, determine how much energy they should purchase at this time.

To make the stochastic control problem tractable, we assume that $e^{(t)}$ also takes values from a finite set as in [1][14], which is denoted by $\mathcal{E}$ and evolves as a finite-state Markov chain with transition probability $\{p_e(e^{(t+1)} \mid e^{(t)})\}$. It has been widely measured in electricity markets that the consumers' demand is significantly influenced by the environment dynamics [17]. To capture this influence, we model the sum demand $d^{(t)}$ as an i.i.d. random variable given the current environment dynamic $e^{(t)}$, with the probability distribution being $\{p_d(d^{(t)} \mid e^{(t)})\}$.

Given the Markovian evolution of $q^{(t)}$, $d^{(t)}$ and $e^{(t)}$, the stochastic control problem can be casted as a Markov Decision Process (MDP). At the beginning of each time slot t, the consumer observes $s^{(t)}$. After this, it purchases an amount $a^{(t)}$ from the electricity producers and uses the purchased (and stored) electricity to operate its appliances. The storage dynamic across time slots is illustrated in Figure 2 and captured by the following expression:

$$b^{(t+1)} = b^{(t)} + a^{(t)} - d^{(t+1)}. \tag{1}$$

The system then evolves into the next time slot $t+1$ with the state $s^{(t+1)}$, with the transition probability expressed as follows:

$$\begin{aligned} &p(s^{(t+1)} \mid s^{(t)}, a^{(t)}) \\ &= p_q(q^{(t+1)} \mid q^{(t)}) p_e(e^{(t+1)} \mid e^{(t)}) \cdot \\ &\quad p_d(b^{(t)} + a^{(t)} - b^{(t+1)} \mid e^{(t)}) \end{aligned} \tag{2}$$

The benefit received by the consumer in each time slot t is determined by the amount of its consumed electricity in this time slot, which equals $\min\{b^{(t)} + a^{(t)}, d^{(t)}\}$, i.e. the maximum amount of electricity consumption cannot exceed the total amount of stored and purchased electricity. Given this, the one-slot utility $u^{(t)}$ received by the consumer in time slot t is expressed as follows:

$$\begin{aligned} u(s^{(t)}, a^{(t)}) = &f(\min\{b^{(t)} + a^{(t)}, d^{(t)}\}) \\ &- c(\max\{b^{(t)} + a^{(t)} - d^{(t)}, 0\}) - q^{(t)} a^{(t)} \end{aligned}. \tag{3}$$

Here $q^{(t)}a^{(t)}$ represents the total price the consumer pays for the purchased electricity and $c(\max\{b^{(t)} + a^{(t)} - d^{(t)}, 0\})$ is the cost incurred for storing the electricity left at the end of time slot t with c being the unit storage cost. $f(\min\{b^{(t)} + a^{(t)}, d^{(t)}\})$ represents the benefit from the electricity consumption and we assume that the form of $f(\cdot)$ is known a priori and satisfies the following conditions:

Assumption 1: $f(x)$ is non-decreasing and concave on x, with $\lim_{x \to \infty} f'(x) = 0$;

Assumption 2: $f(0) = 0$.

Assumption 1 states the fact that a higher consumption leads to a higher benefit for the consumer, whereas the increase on the benefit monotonically decreases against the total consumption. Assumption 2 states the fact that the received benefit is always non-negative. These assumptions are widely adopted in previous works, e.g. [16].

The load scheduling policy of the consumer is a mapping $\pi : \mathcal{Q} \times \mathcal{B} \times \mathcal{E} \to \mathcal{A}$. That is, a policy instructs the action that the consumer takes in each time slot as $a^{(t)} = \pi(s^{(t)})$. The consumer is foresighted and interested in optimizing its policy to maximize its expected long-term utility, which is referred to as the *value function* and defined as follows:

$$U^{\pi}(s^{(0)}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t u(s^{(t)}, a^{(t)}) \mid s^{(0)}\right]. \quad (4)$$

Here $\delta \in [0,1)$ is a constant discount factor, which represents the fact that the consumer puts higher weight on its current utility than its future utility.

Each consumer needs to maximize its long-term utility by solving the following stochastic control problem:

$$\max_{\pi} U^{\pi}(s^{(0)}). \quad (5)$$

Let π^* the optimal solution of the unconstrained MDP (5) and the corresponding optimal long-term utility to be $U^*(s^{(0)})$. It is well-known that π^* and $U^*(s^{(0)})$ can be obtained by recursively solving the following Bellman equation set [18]:

$$U^*(s) = \max_{\pi} \left[\begin{array}{l} f(\min\{b+a,d\}) - c(\max\{b+a-d,0\}) - qa \\ +\delta \sum_{q \in \mathcal{Q}} p_q(q' \mid q) \sum_{e \in \mathcal{E}} p_e(e' \mid e) \\ \quad \sum_{b'=b+a-D}^{b+a} p_d(b+a-b' \mid e) U^*(s') \end{array} \right]. \quad (6)$$

We end this section by summarizing what is known by the consumer at each time slot in order to solve this stochastic control problem. Among all the above variables discussed in this section, the consumer can observe all the state variables, i.e. $q^{(t)}$, $b^{(t)}$, $d^{(t)}$, $e^{(t)}$ as well as its action $a^{(t)}$. However, the consumer cannot observe the state transition probabilities, i.e. $p_e(e^{(t+1)} \mid e^{(t)})$, $p_q(q^{(t+1)} \mid q^{(t)})$, $p_d(d^{(t)} \mid e^{(t)})$. Meanwhile, the electricity prices and the environment dynamics are also unknown by the consumer. All these unknown variables need to be learn by the consumer during the run time.

III. Post-Decision State Based Dynamic Programming

In this section, we analyze and solve the Bellman equation (6) and explore the structural properties of the optimal solution. The traditional algorithms for solving the Bellman equation, e.g. the value iteration and the policy iteration [18], rely on the knowledge of the state transition probabilities, i.e. $p_q(q' \mid q)$, $p_e(e' \mid e)$, and $p_d(b+a-b' \mid e)$, as well as the state space, i.e. $\mathcal{Q}$, D and $\mathcal{B}$. Since these values are not known (or only partially known) a priori, the expectation embedded in (6) cannot be computed using well-known stochastic control techniques. Hence, we propose an online reinforcement learning algorithm to dynamically learn π^* and U^* on-the-fly, without requiring any a priori knowledge of the transition probabilities and the state space.

In the rest of this section, we first introduce the concept of the *post-decision state* in Section III.A, which is an intermediate state of the system in order to capture the known part of the system dynamics. Section III.B then develops a general post-decision state based online learning algorithm that allows the consumer to integrate known information about the system dynamics into its learning process. Exploiting this partially known information dynamically could significantly improve the run-time performance compared to the conventional online learning algorithms, e.g. Q-learning [18]. Finally in Section III.C, we prove the convergence of the proposed algorithm and discuss the structural properties of the optimal policy π^*.

A. Post-Decision State

The key idea behind our proposed learning algorithm is to introduce an intermediate state, which captures the known part of the system dynamics. We call this intermediate state the post-decision state $\tilde{s} \triangleq (\tilde{q}, \tilde{b}, \tilde{e})$. The relationship between a state $s^{(t)}$ and its post-decision state $\tilde{s}^{(t)}$ in time slot t is illustrated in Figure 3. From this figure, it can be noticed that given $s^{(t)} = (q^{(t)}, b^{(t)}, e^{(t)})$, the corresponding post-decision state in the time slot t is computed as follows:

$$\tilde{q}^{(t)} = q^{(t)},\ \tilde{b}^{(t)} = b^{(t)} + a^{(t)},\ \tilde{e}^{(t)} = e^{(t)}. \quad (7)$$

The post-decision state represents the state of the system in each time slot after the consumer performs its action $a^{(t)}$ but before the new demand $d^{(t+1)}$ arrives and the new unit electricity price $q^{(t+1)}$ is set.

Accordingly, we define the post-decision value function $V^*(\tilde{s})$ for a post-decision state $\tilde{s}$ as follows:

$$V^*(\tilde{s}) = \sum_{q \in \mathcal{Q}} p_q(q' \mid q) \sum_{e \in \mathcal{E}} p_e(e' \mid e) \sum_{b'=b+a-D}^{b+a} p_d(b+a-b' \mid e) U^*(s'). \quad (8)$$

For the better illustration, we refer to s as the "normal" state and $U^*(s)$ as the "normal" value function, in order to differentiate with their post-decision counter parts.

By comparing (6) and (8), it can be noticed that the post-decision value function represents the expectation of the consumer's future utilities over the unknown system dynamics. Hence, there is a deterministic mapping from the normal value function to the post-state value function. By substituting (8) into (6), the relationship between the normal

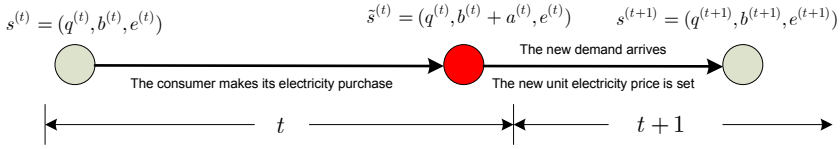

Figure 3 Illustration of post-decision state

value function and the post-state value function can be expressed as follows:

$$U^*(q,b,e) = \max_{a\geq 0}\begin{bmatrix} f(\min\{b+a,d\}) - c(\max\{b+a-d,0\}) \\ -qa + \delta V^*(q,b+a,e) \end{bmatrix}. \quad (9)$$

The above equation shows that the normal value function $U^*(q,b,e)$ at each time slot is obtained from the corresponding post-decision value function $V^*(q,b+a,e)$ at the same time slot by performing the maximization over the action a. Therefore, introducing the post-decision state and the corresponding value functions allows us to factor the state transition probability into known and unknown components and thus optimize the load scheduling policy of the consumer without computing the expected future utility over the unknown system dynamics. In the next section, we discuss propose an online learning algorithm that utilizes adaptive approximation to effectively learn the post-state value functions.

B. Post-Decision State Based Online Learning

We update the post-decision value function using conventional reinforcement learning approaches [18]. In each time slot t, where the post-decision state is $\tilde{q}^{(t)}, \tilde{b}^{(t)}, \tilde{e}^{(t)}$, we update its post-decision value function as follows:

$$V^{(t+1)}(\tilde{q}^{(t)},\tilde{b}^{(t)},\tilde{e}^{(t)}) = (1-\alpha^{(t)})V^{(t)}(\tilde{q}^{(t)},\tilde{b}^{(t)},\tilde{e}^{(t)}) + \alpha^{(t)}U^{(t+1)}(\tilde{q}^{(t)},b^{(t)},\tilde{e}^{(t)}). \quad (10)$$

Here $\alpha^{(t)}$ is the learning rate factor that satisfies $\sum_{t=0}^{\infty}\alpha^{(t)} = \infty$ and $\sum_{t=0}^{\infty}(\alpha^{(t)})^2 < \infty$, e.g. $\alpha^{(t)} = 1/t$. $U^{(t+1)}(\tilde{q}^{(t)},b^{(t)},\tilde{e}^{(t)})$ is the updated normal value function, which is computed as follows:

$$U^{(t+1)}(q,b,e) = \max_{a\geq 0}\begin{bmatrix} f(\min\{b+a,d\}) - c(\max\{b+a-d,0\}) \\ -qa + \delta V^{(t)}(q,b+a,e) \end{bmatrix}. \quad (11)$$

Remark: With (10) and (11), the normal value function and the post-decision value function are updated iteratively in each time slot. In the first step, the normal value function of the current state $s^{(t)} = (q^{(t)},b^{(t)},e^{(t)})$ is updated to $U^{(t+1)}(q^{(t)},b^{(t)},e^{(t)})$ using the post-decision value function $V^{(t)}(q^{(t)},b^{(t)}+a,e^{(t)})$. In the second step, the post-decision value function of the current post-decision state $\tilde{s} = (q^{(t)},b^{(t)}+a^{(t)},e^{(t)})$ is updated to $V^{(t+1)}(q^{(t)},b^{(t)}+a^{(t)},e^{(t)})$ using the updated normal value function $U^{(t+1)}(q^{(t)},b^{(t)},e^{(t)})$. In the next section, we prove that such iterative update process introduced by (10) and (11) ensures both the normal value function and the post-decision value function converge to their optimal values, i.e. the solution of (6).

The above iterative update process introduced by (10) and (11), though ensures the convergence to the optimal value, only updates the post-state value function of the currently visited post-decision state. Nevertheless, it should be noted that the temporal transition of the unit electricity price q, the environment dynamics e and the consumer demand d are all independent to the consumer's action a. Therefore, instead of updating the post-decision value function only for the state $\tilde{s}^{(t)} = (q^{(t)},b^{(t)}+a^{(t)},e^{(t)})$, we can perform a batch update in time slot t for the post-decision value function at any state $\tilde{s} = (\tilde{q},\tilde{b},\tilde{e})$ such that $\tilde{q} = q^{(t)}$ and $\tilde{e} = e^{(t)}$, which is shown as follows:

$$V^{(t+1)}(\tilde{q}^{(t)},\tilde{b},\tilde{e}^{(t)}) = (1-\alpha^{(t)})V^{(t)}(\tilde{q}^{(t)},\tilde{b},\tilde{e}^{(t)}) + \alpha^{(t)}U^{(t+1)}(\tilde{q}^{(t)},\tilde{b}-a^{(t)},\tilde{e}^{(t)}),\ \forall\tilde{b} \quad (12)$$

With the batch update (12), we are able to update all the post-decision states $\{(q^{(t)},\tilde{b},e^{(t)}),\forall\tilde{b}\}$ and hence, the convergence speed of our proposed learning algorithm is significantly improved, which will be shown in Section IV.

In summary, our proposed online learning algorithm based on post-decision state is illustrated in Table 2.

C. Structure of the Optimal Policy

In this section, we analyze the structure of the optimal policy as well as the associated optimal normal and post-decision value functions, which are computed by Algorithm 1.

First we analyze the convergence property of the optimal policy, which is proven in the following theorem.

Theorem 1. The post-decision state based online learning algorithm converges to the optimal post-decision value function $\{V^*(\tilde{s})\}$ when the sequence of learning rates $\alpha^{(n)}$

Initialize: $V^{(0)}(\tilde{s}) = 0$ for all $\tilde{s}$; $\tilde{s}^{(0)} = (q^{(0)},b^{(0)},e^{(0)})$; $\tilde{s}^{(0)} = (q^{(0)},b^{(0)},e^{(0)})$; $t = 1$

Repeat

(1) Observe the customer demand $d^{(t)}$ and the unit electricity price $q^{(t)}$;

(2) Update the normal state $s^{(t)} = (q^{(t)},b^{(t)},e^{(t)})$ with $b^{(t)} = \tilde{b}^{(t)} - d^{(t)}$;

(3) Batch update the post-decision value functions as in (12);

(4) Compute the optimal action $a^{(t)}$ for the current normal state $s^{(t)}$ as in (11);

(5) Update the post-decision state $\tilde{s}^{(t)} = (q^{(t)},\tilde{b}^{(t)},e^{(t)})$ with $\tilde{b}^{(t)} = b^{(t)} + a^{(t)}$;

(6) $t := t+1$

End

Table 2 PDS-based load scheduling algorithm:

satisfies $\sum_{t=0}^{\infty}\alpha^{(t)}=\infty$ and $\sum_{t=0}^{\infty}(\alpha^{(t)})^2<\infty$.

Proof: From (10) and (11), it is known that the PDS-based load scheduling algorithm defined in Table 2 can be written using the following recursion:

$$V^{(t+1)}(\tilde{q}^{(t)},\tilde{b}^{(t)},\tilde{e}^{(t)})=(1-\alpha^{(t)})V^{(t)}(\tilde{q}^{(t)},\tilde{b}^{(t)},\tilde{e}^{(t)}) + \alpha^{(t)}\max_{a\geq 0}\begin{bmatrix} f(\min\{\tilde{b}^{(t)},d^{(t)}\})-c(\max\{\tilde{b}^{(t)}-d^{(t)},0\}) \\ -\tilde{q}^{(t)}a+\delta V^{(t)}(\tilde{q}^{(t)},b^{(t)},\tilde{e}^{(t)}) \end{bmatrix} . \quad (13)$$

Let $h:\mathbb{R}^{|\mathcal{S}|}\rightarrow\mathbb{R}^{|\mathcal{S}|}$ be a map such that $h(V)=\left[h_{\tilde{s}}(V)\right]_{\tilde{s}}$ with

$$h_{\tilde{s}}(V)=\max_{a\geq 0}\begin{bmatrix} f(\min\{\tilde{b},d\})-c(\max\{\tilde{b}-d,0\}) \\ -\tilde{q}a+\delta V(\tilde{q},b,\tilde{e}) \end{bmatrix} - V(\tilde{q},\tilde{b},\tilde{e}), \quad (14)$$

and let $F:\mathbb{R}^{|\mathcal{S}|}\rightarrow\mathbb{R}^{|\mathcal{S}|}$ be a map such that $F(V)=\left[F_{\tilde{s}}(V)\right]_{\tilde{s}}$ with

$$h_{\tilde{s}}(V)=\max_{a\geq 0}\begin{bmatrix} f(\min\{\tilde{b},d\})+\delta V(\tilde{q},b,\tilde{e}) \\ -\tilde{q}a-c(\max\{\tilde{b}-d,0\}) \end{bmatrix}. \quad (15)$$

Then $h(V)=F(V)-V$. As in [19], it can be shown that the convergence of our proposed algorithm is equivalent to the convergence of the associated O.D.E.:

$$\dot{V}=F(V)-V:=h(V). \quad (16)$$

Since the map $F:\mathbb{R}^{|\mathcal{S}|}\rightarrow\mathbb{R}^{|\mathcal{S}|}$ is a maximum norm δ-contraction [22], the asymptotic stability of the unique equilibrium point of the above O.D.E. is guaranteed in [19]. This unique equilibrium point corresponds to the optimal post-decision state value function $\{V^*(\tilde{s})\}$. ■

Theorem 1 shows that the online learning algorithm is able to learn the optimal post-decision value function $\{V^*(\tilde{s})\}$ in the long run. Since the optimal normal value function $\{U^*(s)\}$ is a deterministic function of $\{V^*(\tilde{s})\}$, it can be concluded that our proposed online learning algorithm also converges to $\{U^*(s)\}$ and thus the optimal policy π^*. Therefore, we prove that the consumer is able to learn the optimal load scheduling policy through the load scheduling algorithm proposed in Table 2.

In the rest of this section, we characterize the structural properties of the optimal policy π^*. Understanding these properties can significantly reduce the size of the space over which π is optimized and thus greatly simplify the complexity of solving the equation (11) (i.e. Step (4) in Table 2).

First, we show in the following proposition that in the optimal policy π^*, the amount of purchased electricity in each time slot is always upper-bounded.

Proposition 1. There is a constant A such that $\pi^*(s)\leq A,\forall s\in\mathcal{S}$.

Proof: This can be straightforwardly obtained by taking the first order derivative over the one-slot utility function (3) over a. It can be observed at each state s, there is a threshold value A_s such that $\frac{\partial u(s)}{\partial a}<0$ when $a>A_s$. Since the long-term utility (4) is a linear combination of the one-slot utilities. We have $A=\max_{s\in\mathcal{S}}\{A_s\}$. ■

Proposition 1 proves that the consumer's purchased amount of electricity in one time slot will not be arbitrarily large but is always upper-bounded by a constant value A. This is due to the fact that the benefit function $f(\cdot)$ is concave while the storage cost linearly increases against the purchased amount a. Therefore, when a is too large, the one-slot utility $u(s^{(t)},a^{(t)})$ and thus the long-term utility $U(s)$ start to decrease. With Proposition 1, it can be determined that the consumer always chooses its action from a finite action space, i.e. $a\in[0,A]$. This result guarantees that the MDP proposed in this paper can be solved within a finite state space and a finite action space, which ensures the proposed online learning algorithm to be feasible in practical systems.

The next proposition reveals the monotonicity properties embedded in π^*. By exploiting such monotonicity properties, the optimization in (11) can be further simplified.

Proposition 2. (i) Given two unit electricity prices $q_1<q_2$, $\pi^*(q_1,b,e)\geq\pi^*(q_2,b,e),\forall b,e$ always holds;

(ii) Given $b_1<b_2$, $\pi^*(q,b_1,e)\geq\pi^*(q,b_2,e),\forall q,e$.

Proof: Suppose that $\pi^*(q_1,b,e)<\pi^*(q_2,b,e)$ for some b,e and some $q_1<q_2$. Then according to the recursive equation (6), it is obvious that we can always find an action $a\neq\pi^*(q_1,b,e)$ at the state $s=(q_1,b,e)$, such that

$$U^*(s)<f(\min\{b+a,d\})-c(\max\{b+a-d,0\})-q_1a +\delta\sum_{q\in\mathcal{Q}}p_q(q'\mid q_1)\sum_{e\in\mathcal{E}}p_e(e'\mid e)\sum_{b'=b+a-D}^{b+a}p_d(b+a-b'\mid e)U^*(s'). \quad (17)$$

This contradicts the fact that $\pi^*(q_1,b,e)$ is the optimal action at the state $s=(q_1,b,e)$. Hence, Statement (i) is proven. Statement (ii) can be proven in a similar manner. ■

Statement (i) in Proposition 2 proves that given b,e, the purchased amount in each time slot monotonically decreases with the unit electricity price q. This is straightforward since a lower price gives the consumer a higher incentive to purchase and store more electricity in the current time slot in order to reduce its future cost on electricity purchase. Similarly, statement (ii) proves that for a given electricity price q, the consumer has less incentive to purchase electricity when it has a larger storage. Proposition 2 provides further refinement on the (feasible) action space of the load scheduling.

	Value iteration	RTDP	Q-learning	PDS learning
Average utility	1.4054	0.9047	0.8167	1.3394
Average consumed energy	0.7948 kWh	0.4537 kWh	0.4272 kWh	0.5847 kWh
Average purchased price	0.2018/kWh	0.2930/kWh	0.2928/kWh	0.2514/kWh

Table 3 Performences achieved by different algorithms per slot

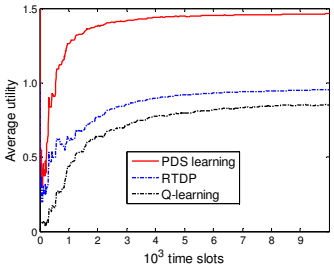

Figure 4 Run-time performances of different learning algorithms

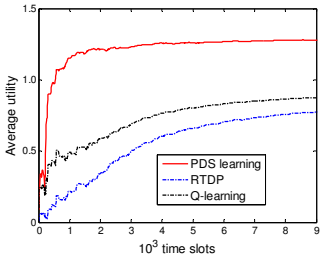

Figure 5 Run-time performances of different learning algorithms when the price evolution is non-Markovian

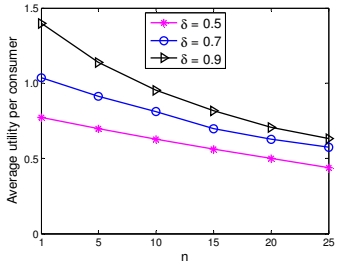

Figure 6 Impact of group size under PDS learning

IV. Illustrative Results

In this section, we provide numerical results to illustrate the performance of our proposed load scheduling algorithm. We consider a power grid with 100 consumers, where the length of each time slot is 1 hour. The environment dynamics e represents the hour that each time slot is located in a day and hence, we have $\mathcal{E} = \{0,1,...,23\}$ and

$$e^{(t)} = \mathrm{mod}(t,24). \quad (18)$$

In one day, we divide the time slots into peak and non-peak hours. Specifically, the peak hours are 6pm to 12am and the rest hours are non-peak hours. The demand of each consumer in each time slot t follows a truncated Gaussian distribution in the region $[0,2.5kWh]$, given $e^{(t)}$. Specifically, we assume that

$$p_d(d^{(t)} \mid e^{(t)}) = \begin{cases} \mathcal{N}(0.5,0.2^2), & if\ e^{(t)} \in [0,17] \\ \mathcal{N}(1,0.1^2), & if\ e^{(t)} \in [18,23] \end{cases}. \quad (19)$$

The unit electricity price is taken from a finite set $\mathcal{Q} = \{0.1,0.2,...,0.5\}$. We also set $c = 0.1$ and the benefit function to be a logarithmic function as in [14]:

$$f(x) = \log(1+x). \quad (20)$$

We first evaluate the performance of our proposed PDS-based load scheduling algorithm and compare it with three benchmark algorithms:

(1) Value iteration [18] is an off-line algorithm. It requires the full knowledge of the underlying MDP, including the state space, the action space, and the state transition probability. Also, the computation complexity of value iteration is usually significantly higher than online reinforcement learning methods. However, this algorithm is ensured to converge to the optimal solution and hence, we use it as the optimal benchmark.

(2) Q-learning [18] is a model-free reinforcement learning algorithm. It does not require a priori knowledge of the underlying MDP, but usually suffers from slow convergence speed.

(3) Real-time dynamic programming (RTDP) [21] is a model-based online learning algorithm. When implementing RTDP, the learning agent first constructs a statistic model of the underlying MDP and then updates the state transition probabilities in this statistic model using its past experiences. Therefore, the state space of the MDP needs to be known a priori.

Table 3 shows the average performance received by an individual consumer when running the four algorithms separately in the considered power grid [3]. Here the "PDS learning" refers to our proposed PDS-based load scheduling algorithm. It can be noticed that our algorithm significantly outperforms the other two online learning algorithms, with on average a higher consumed electricity per slot and a lower price for the electricity purchase. Consequently, the average one-slot utility achieved by our algorithm is close to the optimal value obtained with the off-line value iteration algorithm.

Figure 4 further illustrates the run-time performances of the online learning algorithms across 10000 time slots. It can be observed that PDS learning converges after 1122 time slots (with the run-time average utility achieving 90% of the highest value), while RTDP converging after 2983 time slots and Q-learning converging after 3132 time slots. Also, the average one-slot utilities achieved by RTDP and Q-learning upon convergence are significantly worse than that achieved by PDS, which indicates that both RTDP and Q-learning are not able to learn the optimal load scheduling policy.

So far, the experiments are all conducted under the assumption that the system dynamics evolve following an MDP model. In the next experiment, we evaluate how our proposed algorithm performs when the system dynamics

[3] For the online algorithms (PDS learning, Q-learning, RTDP), we run each of them for 10000 time slots. For the off-line value iteration, we run the algorithm until it converges (because there is no intermediate output for the value iteration before its convergence).

evolve in a non-Markovian way. Specifically, we assume that the unit electricity price in each time slot is determined by the following dynamics:

$$p_q(q^{(t)} \mid q^{(t-1)}, q^{(t-2)}) = \begin{cases} 0.5, & if\ q^{(t)} = q^{(t-1)} \\ 0.3, & if\ q^{(t)} = q^{(t-2)} \\ 0.2, & otherwise \end{cases}. \quad (21)$$

Figure 5 plots the performances of different online learning algorithms under such non-Markovian pricing, while the other settings remain unchanged. It can be observed that due to its fast learning speed, the PDS learning can still quickly catch up with the non-Markovian dynamics on electricity prices with its run-time performance remaining unaffected. Nevertheless, the RTDP and the Q-learning converge significantly slower. Therefore, this experiment shows that our proposed PDS learning algorithm, though proposed based on a Markovian model, is still able to perform well in non-Markovian settings.

In the final experiment, we examine how the PDS learning performs in the scenario where the load scheduling is performed collaboratively by a group of consumers with the help of an aggregator. Figure 6 shows how the average one-slot utility received by one consumer changes against the group size (i.e. the number of consumers in the group). It is interesting to observe that the average one-slot utility always monotonically decreases against the group size n. To explain this phenomenon, we first define the sum load demand of the group in each time slot n as $d_{sum}^{(t)} = \sum_{i=1}^{n} d_i^{(t)}$. According to (19), it is easy to know that the variance of $d_{sum}^{(t)}$ monotonically increases against n. Therefore as n increases, the sum load demand in each time slot becomes more difficult to predict due to its larger variance, which in turn slows down the convergence speed of the PDS learning and thus reduces its performance. Hence, it is not always beneficial to perform such collaborative load scheduling compared to the individual load scheduling.

V. CONCLUSION

In this paper, we propose a novel price-based load scheduling algorithm using electric storage devices. Our algorithm is able to learn the optimal load scheduling policy without requiring any a priori knowledge of the system dynamics. By introducing the post-decision state and batch update, we prove that our proposed algorithm provides significantly faster convergence speed and thus the run-time performance is at least 30% better compared to the state-of-the-art foresighted scheduling algorithms.

REFERENCE

[1] M. Alizadeh, X. Li, Z. Wang, A. Scaglione, and R. Melton, "Demand-Side Management in the Smart Grid: Information Processing for the Power Switch," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 55 – 67, 2012.

[2] S. Borenstein, "The Long-run Efficiency of Real-time Electricity Pricing," *Energy J.*, vol. 26, no. 3, pp. 93 – 116, 2005.

[3] K. Nygard, S. Ghosn, M. Chowhury, D. Loegering, R. McCulloch, and P. Ranganathan, "Optimization Models for Energy Reallocation in a Smart Grid," *IEEE INFOCOM Workshops*, 2011.

[4] K. Clement-Nyns, E. Haesen, and J. Driesen, "The Impact of Charging Plug-in Hybrid Electric Vehicles on a Residential Distribution Grid," *IEEE Trans. on Power Systems*, vol. 25, no. 1, pp. 371 – 380, 2010.

[5] C. Su and D. Kirschen, "Quantifying the Effect of Demand Response on Electricity Markets," *IEEE Trans. on Power Systems*, vol. 24, no. 3, pp. 1199 – 1207, 2009.

[6] H. Song, C. Liu, J. Lawarree, and R. Dahlgren, "Optimal Electricity Supply Bidding by Markov Decision Process," *IEEE Trans. on Power Systems*, vol. 15, no. 2, pp. 618 – 624, 2000.

[7] A. Lam, L. Huang, A. Silva, and W. Saad, "A Multi-layer Market for Vehicle-to-Grid Energy Trading in the Smart Grid" *IEEE INFOCOM Workshop*, 2012..

[8] K. Herter, "Residential Implementation of Critical Peak Pricing of Electricity," *Energy Policy*, vol. 35, pp. 2121 – 2130, 2007.

[9] M. He, S. Murugesan, and J. Zhang, "Multiple Timescale Dispatch and Scheduling for Stochastic Reliability in Smart Grids with Wind Generation Integration," *IEEE INFOCOM*, 2011

[10] B. Kim, S. Ren, M. van der Schaar, and J. Lee, "Bidirectional Energy Trading for Residential Load Scheduling and Electric Vehicles," *IEEE INFOCOM*, 2013.

[11] 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, 2010.

[12] D. Bunn, "Forecasting Loads and Prices in Competitive Power Markets," *Proc. IEEE*, vol. 88, pp. 163 – 169, 2000.

[13] I. Koutsopoulos, V. Hatzi, and L. Tasiulas, "Optimal Energy Storage Control Policies for the Smart Power Grid," *IEEE SmartGridComm*, 2012

[14] P. Reddy and M. Veloso, "Strategy Learning for Autonomous Agents in Smart Grid Markets," *Int'l Joint Conf. on Artificial Intelligence*, 2011.

[15] M. Alizadeh, A. Scaglione, and R. Thomas, "From Packet to Power Switching: Digital Direct Load Scheduling," *IEEE J. on Sel. Areas in Commun.*, vol. 30, no. 6, pp. 1027 – 1036, 2012.

[16] P. Vytelingum, T. Voice, S. Ramchurn, A. Rogers, and N. Jennings, "Agent-based Micro-storage Management for the Smart Grid," *Int'l Conf. on Autonomous Agents and Multiagent Systems* (AAMAS), 2010.

[17] S. Mirasgedis, Y. Sarafidis, E. Georgopoulous, D. Lalas, M. Moschovits, F. Karagiannis, and D. Papakonstantinous, "Models for mid-term electricity demand forecasting incorporating weather influences," *Energy*, vol. 31, no. 2 – 3, pp. 208 – 227, 2006.

[18] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," *MIT Press*, 1998.

[19] V. Borkar and S. Meyn, "The ODE Method for Convergence of Stochastic Approximation and Reinforcement learning," *SIAM J. Control Optimization*, vol. 28, pp. 447 – 469, 1999.

[20] F. Fu and M. van der Schaar, "Structure-Aware Stochastic Control for Transmission Scheduling," *IEEE Trans. on Vehicular Tech.*, vol. 61, no. 9, pp. 3931 – 3945, 2010.

[21] Y. Zhang, F. Fu, and M. van der Schaar, "On-Line Learning and Optimization for Wireless Video Transmission," *IEEE Trans. on Signal Process.*, vol. 58, no. 6, pp. 3108 – 3124, 2010.

[22] D. Bertsekas, "Dynamic Programming and Optimal Control," *Athena Scientific*, 2005.