Real-Time Pricing for Demand Response Based on Stochastic Approximation

Pedram Samadi, *Student Member, IEEE*, Hamed Mohsenian-Rad, *Member, IEEE*, Vincent W.S. Wong, *Senior Member, IEEE*, and Robert Schober, *Fellow, IEEE*

***Abstract*—In this paper, we propose a new pricing algorithm to minimize the peak-to-average ratio (PAR) in aggregate load demand. The key challenge that we seek to address is the energy provider's uncertainty about the impact of prices on users' load profiles, in particular when users are equipped with automated *energy consumption scheduling* (ECS) devices. We use an iterative *stochastic approximation* approach to design two real-time pricing algorithms based on *finite-difference* and *simultaneous perturbation* methods, respectively. We also propose the use of a *system simulator unit* (SSU) that employs approximate dynamic programming to simulate the operation of the ECS devices and users' price-responsiveness. Simulation results show that our proposed real-time pricing algorithms reduce the PAR in aggregate load and help the users to reduce their energy expenses.**

***Keywords*: Demand response, real-time pricing, PAR minimization, stochastic approximation, simultaneous perturbation.**

I. Introduction

Demand response (DR) is an important component of the future smart grid [1]–[3]. *Direct load control* (DLC) and *price-based load control* are two general categories of DR programs. In DLC programs, based on a contract between the energy provider and the users, the energy provider can remotely control the operation and energy consumption of certain appliances for users [4]–[10]. In contrast, in price-based programs, the energy provider provides economic incentives to consumers by changing the electricity price for different times of a day such that users are encouraged to shift their usage of high-power appliances to off-peak hours [11]–[18].

With the growing deployment of *advanced metering infrastructure* (AMI) [19] and automated energy consumption scheduling (ECS) devices [20]–[25], real-time pricing (RTP) is gradually becoming a feasible DR solution. In general, it is difficult for power users to follow the RTP price variations to make appropriate decisions accordingly. In this regard, ECS devices can help by making such *price-responsive* decisions on behalf of users to achieve certain objectives. Examples of such objectives include minimizing the energy expenses [20], maximizing the social welfare [21]–[25], minimizing both the energy expenses and the waiting time [26], and maintaining system stability with minimum curtailment [27]. However, while the use of ECS devices improves users' *rationality* in response to price changes, such ECS devices can also introduce new DR challenges such as *load synchronization* [26] and price instability [28], [29]. Therefore, the effect of the automated ECS devices on users' price-responsiveness is not obvious and yet to be investigated. It has been shown that load synchronization can be avoided by adopting pricing tariffs with *inclining block rates* (IBRs), where the marginal price increases when the load increases [26].

In this paper, we address minimizing the *peak-to-average ratio* (PAR) in the aggregate load demand through pricing under the practical scenario that the utility is uncertain about users' price-responsiveness. While we assume that users are equipped with ECS devices, our approach is quite different from all prior works, e.g., in [20]–[27], which do not take into account the uncertainty in users' price-responsiveness. Note that such uncertainty is inevitable to preserve *user privacy* [30].

Some related literature can be summarized as follows. In [24], Chen *et al*. devised a Stackelberg game approach in which the energy provider acts as a leader and users are followers. This design intends to jointly maximize the social welfare of all users and the revenue of the energy provider. The algorithm in [24] requires detailed information about users' energy consumption needs. However, for the scheme proposed in this paper, users are not required to submit their demand requirements at the beginning of the operation period. As a result, our design is more practical and preserves users' privacy. The authors of [20] proposed a game theoretic approach to minimize the PAR of the system, where users interact with each other and change their power consumption accordingly. However, such interactions may take a long time to converge, in particular in the presence of a large number of users. In contrast, our design does not involve direct user interaction. Therefore, it converges much faster. Finally, while in [31], the authors devised a method which takes into account the *load uncertainty* to minimize the energy payment for each user, here, our focus is on designing a pricing algorithm and our design objective is to minimize the PAR in the aggregate load.

The block diagram of our proposed real-time pricing model is shown in Fig. 1. Our main contributions are as follows:

- We propose two iterative algorithms to be implemented in a *price control unit* (PCU) for minimizing the PAR of the aggregate load based on the information provided by the *system simulator unit* (SSU). The first algorithm, called

Manuscript was received on Mar. 4, 2013, revised on Jun. 6, 2013 and Sept. 15, 2013, and accepted on Nov. 2, 2013. This work was supported in part by a Strategic Project Grant (STPGP-396838) from the Natural Science and Engineering Research Council of Canada (NSERC). P. Samadi, V.W.S. Wong, and R. Schober are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, V6T 1Z4, email: {psamadi, vincentw, rschober}@ece.ubc.ca. H. Mohsenian-Rad is with the Department of Electrical Engineering, University of California, Riverside, CA, USA, email: hamed@ee.ucr.edu

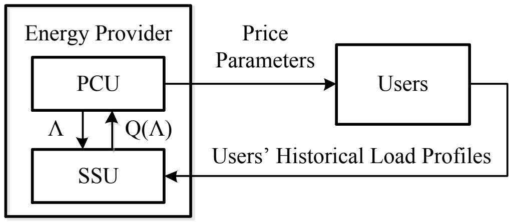


Fig. 1. The block diagram of the proposed closed-loop pricing model.

finite-difference price selection (FDPS), uses a variation of the finite-difference technique [32] to approximate the gradient of the PAR minimization objective function by making small one-at-a-time changes to each individual element of the input price parameter vector. The FDPS algorithm requires only few iterations for convergence. However, it needs a large number of measurements of the objective function in each iteration.

- The second algorithm, called simultaneous perturbation price selection (SPPS), is based on the simultaneous perturbation technique [32]. Unlike the FDPS algorithm, all elements of the input variable are jointly and randomly perturbed to approximate the gradient. As a result, the SPPS algorithm significantly reduces the number of measurements of the objective function in each iteration, compared to the FDPS algorithm. Yet, we show that it achieves a similar performance.
- We propose an approximate dynamic programming scheme which simulates how users automatically respond to various price values to eliminate the need for users to reveal their detailed energy consumption needs to the energy provider. This assures user privacy.
- Simulation results show that our proposed pricing algorithms reduce the PAR in aggregate load. In addition, we show that adopting the new pricing algorithms is also beneficial for the users if they are equipped with automated control units.

The rest of this paper is organized as follows. The system model and problem formulation are introduced in Section II. The FDPS and SPPS algorithms are developed in Section III. The SSU is explained in Section IV. Simulation results are presented in Section V. The paper is concluded in Section VI.

II. Problem Formulation

Let $\mathcal{U}$ denote the set of all users. Let $\mathcal{A}_u$ denote the set of all appliances of user $u \in \mathcal{U}$. We denote $\mathcal{M}_u$ as the set of must-run appliances of user u, $\mathcal{C}_u$ as the set of controllable appliances of user u, and $\mathcal{N}_u$ as the non-interruptible subset of $\mathcal{C}_u$. For each user, we assume that there is an ECS unit which is embedded in the user's smart meter and controls the user's power consumption [16], [17]. The users' responses to the price changes are done automatically using the ECS units. All ECS units are connected to the energy provider through a two-way communication infrastructure. The operation period is divided into T time slots. We define binary variable $x^t_{u,a} \in \{0,1\}$ as the state of power consumption of appliance $a \in \mathcal{A}_u$ at time slot $t \in \mathcal{T}$, where $\mathcal{T} \triangleq \{1,\ldots,T\}$. We set $x^t_{u,a} = 1$ if appliance a operates in time slot t; otherwise, we have $x^t_{u,a} = 0$. For each user u, $E_{u,a}$ is the total energy requirement of appliance $a \in \mathcal{A}_u$, $\gamma_{u,a}$ is the nominal power consumption of appliance a, and $T_{u,a} = E_{u,a}/\gamma_{u,a}$.

A. Centralized Load Control Algorithm

Assuming that the energy provider is aware of all users' energy needs and is capable of remotely controlling the ECS devices of all users, the centralized load control problem to minimize the PAR in aggregate load can be formulated as

$$\underset{\mathbf{x}_{u,a}\,\in\,\tilde{\mathcal{X}}_u,\ \forall\, a\,\in\,\mathcal{A}_u,\ \forall\, u\,\in\,\mathcal{U}}{\text{minimize}} \quad \frac{T\max\{L_1,\ldots,L_T\}}{\sum_{t=1}^{T} L_t}, \tag{1}$$

where

$$\mathbf{x}_{u,a} \triangleq (x^1_{u,a},\ldots,x^T_{u,a}), \tag{2}$$

$$L_t = \sum_{u\in\mathcal{U}}\sum_{a\in\mathcal{A}_u}\gamma_{u,a}x^t_{u,a}, \tag{3}$$

and the feasible set $\tilde{\mathcal{X}}_u$ is defined as

$$\tilde{\mathcal{X}}_u = \Bigg\{\mathbf{x}_{u,a} \,\Bigg|\, x^t_{u,a}\in\{0,1\},\ \forall a\in\mathcal{A}_u,\ \forall\, t\in\mathcal{T},$$
$$\gamma_{u,a}\sum_{t=\alpha_{u,a}}^{\beta_{u,a}} x^t_{u,a} = E_{u,a},\ \forall\, a\in\mathcal{C}_u,$$
$$\gamma_{u,a}\sum_{t=\alpha_{u,a}}^{\alpha_{u,a}+T_{u,a}-1} x^t_{u,a} = E_{u,a},\ \forall\, a\in\mathcal{M}_u,$$
$$x^t_{u,a} = 1,\ \forall\, a\in\mathcal{N}_u, \forall\, t\in\mathcal{T}_{u,a}, 0 < E^t_{u,a} < E_{u,a}\Bigg\}. \tag{4}$$

Here, $\alpha_{u,a}$ is the earliest time at which the operation of appliance a could be scheduled, $\beta_{u,a}$ is the deadline by which the operation of appliance a should be finished, $\mathcal{T}_{u,a} = \{\alpha_{u,a},\ldots,\beta_{u,a}\}$, and $E^t_{u,a}$ is the amount of energy required to finish the operation of appliance $a \in \mathcal{A}_u$ while the system is at time slot t and is calculated as

$$E^t_{u,a} = \left[E_{u,a} - \gamma_{u,a}\sum_{k=1}^{t-1} x^k_{u,a}\right]^+. \tag{5}$$

The first constraint in (4) indicates that each appliance can be either *on* or *off*. The second constraint implies that the operation of each appliance should be scheduled within its feasible interval. The third constraint indicates that the operation of must-run appliances should be started immediately. The last constraint guarantees that the operation of non-interruptible appliances will continue, once they become active.

The ECS unit does not change the total load of the users, $\sum_{t=1}^{T} L_t$, and the denominator in (1) is a constant. Thus, we introduce an auxiliary variable Γ and rewrite problem (1) as

$$\begin{aligned} &\underset{\substack{\Gamma,\,\mathbf{x}_{u,a}\,\in\,\tilde{\mathcal{X}}_u,\\ \forall\, a\,\in\,\mathcal{A}_u,\,\forall\, u\,\in\,\mathcal{U}}}{\text{minimize}} && \Gamma \\ &\text{subject to} && \sum_{u\in\mathcal{U}}\sum_{a\in\mathcal{A}_u}\gamma_{u,a}x^t_{u,a} \le \Gamma,\quad \forall\, t\in\mathcal{T}. \end{aligned} \tag{6}$$

Problem (6) is a linear mixed-integer program and can be solved using software such as MOSEK [33]. Its solution provides a performance benchmark for any load control algorithm that minimizes the PAR of the aggregate load while satisfying the demand requirements of all users.

B. Decentralized Price-Based Load Control Algorithm

In this section, we assume that the energy provider has no control over users' behavior and it may only influence the load by changing the price parameters. Recall from Section I that RTP and IBR are two non-flat pricing models that are used to encourage consumers to shift some of their load from peak hours to off-peak hours and also to prevent load synchronization.

Let $L_u^t \triangleq \sum_{a\in\mathcal{A}_u} \gamma_{u,a} x_{u,a}^t$ denote the total power consumption of user u at time slot t. Let $\lambda_t(L_u^t)$ denote the selected price of electricity in time slot t as a function of the user's power consumption in that time slot. By combining RTP and IBR [34], [35], the price function $\lambda_t(L_u^t)$ is defined as [26]:

$$\lambda_t(L_u^t) = \begin{cases} m_t, & \text{if } 0 \le L_u^t \le b_t, \\ n_t, & \text{if } L_u^t > b_t, \end{cases} \tag{7}$$

where m_t, n_t, and b_t are price parameters, and $m_t \le n_t$. Also, let $\Lambda_t \triangleq (m_t, n_t, b_t)$ and $\Lambda \triangleq (\Lambda_1, \dots, \Lambda_T)$. The general pricing function in (7) represents an RTP structure that is combined with IBR. Based on this combined model, the price of electricity depends on the *time of day* and also the *total load*. We assume that the price parameters for each time slot are selected such that the PAR of the aggregate load in the system is minimized. Thus, the best choice for the price parameters in each time slot is that obtained by solving the following optimization problem:

$$\begin{aligned} \underset{\Lambda}{\text{minimize}} \quad & Q(\Lambda) \qquad (8) \\ \text{subject to} \quad & m_t^{min} \le m_t \le m_t^{max},\ \forall\, t \in \mathcal{T}, \\ & n_t^{min} \le n_t \le n_t^{max},\ \forall\, t \in \mathcal{T}, \\ & b_t^{min} \le b_t \le b_t^{max},\ \forall\, t \in \mathcal{T}, \\ & m_t \le n_t,\ \forall\, t \in \mathcal{T}, \end{aligned}$$

where

$$Q(\Lambda) = \max\{L_1(\Lambda), \dots, L_T(\Lambda)\}, \tag{9}$$

$L_t(\Lambda)$ denotes the aggregate load at time slot t that depends on the price parameters. m_t^{min}, m_t^{max}, n_t^{min}, n_t^{max}, b_t^{min}, and b_t^{max} are lower and upper bounds for the price parameters m_t, n_t, and b_t, respectively. To devise an efficient pricing algorithm capable of minimizing the PAR, *the energy provider needs to know the behavior of the users in response to the selected price parameters.* With the deployment of the ECS devices, it is very challenging to anticipate the response of the users to different price parameters. Therefore, we *cannot* have an explicit expression for $Q(\Lambda)$ and consequently it is *not* possible to obtain a closed-form analytical solution for (8).

Due to the challenges explained above, we propose to solve problem (8) using an iterative algorithm that does not require a closed-form expression for $Q(\Lambda)$. That is, we follow a step-by-step procedure that moves from an initial guess to a final value which is close to the optimum solution of problem (8). There exist different methods to solve problem (8) iteratively. In this regard, we propose to equip the energy provider with an SSU, as shown in Fig. 1, that simulates the likely behavior of the users in response to price parameters announced by the energy provider. The information produced by the SSU will then be used by the PCU to select prices.

III. Price Control Unit (PCU)

Recall from Section II-B that finding a closed-form solution for problem (8) is challenging. An alternative is an iterative algorithm using a gradient method. In this regard, we need to *approximate* the gradient from noisy measurements of $Q(\Lambda)$. Next, we propose two different methods for this purpose.

A. Finite-Difference Price Selection (FDPS)

Using the finite-difference technique [32, Ch. 6], the gradient of the objective function can be approximated by making small one-at-a-time changes to each of the individual elements of Λ. That is, the jth element of vector Λ is perturbed and the changes in the objective function are measured. The ratio of the changes in the objective function to the amount of the perturbation of the jth element of vector Λ approximates the jth element of the gradient vector of objective function $Q(\Lambda)$. The general recursive procedure of updating the price parameters in each time slot can be written as

$$\Lambda^{i+1} = \Lambda^i - \sigma^i \hat{g}^i(\Lambda^i), \tag{10}$$

where $p \times 1$ column vector $\hat{g}^i(\Lambda^i)$ is an estimate of the gradient of $Q(\Lambda)$, $\nabla Q(\Lambda)$, at iteration i based on the measurements of $Q(\Lambda)$[1], Λ^i is the input vector Λ at iteration i, and $p = 3T$ is the size of vector Λ^i. The step size $\sigma^i > 0$ is reduced as the number of iterations increases to assure convergence. In our proposed FDPS algorithm, we use one-sided gradient approximations which involve evaluations of the form $Q(\Lambda^i + \text{perturbation})$ and $Q(\Lambda^i)$. That is, we obtain the gradient estimate as

$$\hat{g}^i(\Lambda^i) = \begin{bmatrix} \frac{Q(\Lambda^i + c^i\zeta_1) - Q(\Lambda^i)}{c^i} \\ \vdots \\ \frac{Q(\Lambda^i + c^i\zeta_p) - Q(\Lambda^i)}{c^i} \end{bmatrix}, \tag{11}$$

where ζ_j denotes a $p \times 1$ vector with a 1 in the jth position and zeros elsewhere, and $c^i > 0$ is the magnitude of the perturbations. Among different methods proposed for selecting coefficients σ^i and c^i, some specific forms have been suggested in practice which also satisfy the conditions required for convergence of the algorithm [32, Ch. 6]:

$$\sigma^i = \frac{\sigma}{(i+1+A)^\alpha}, \quad c^i = \frac{c}{(i+1)^\gamma}, \tag{12}$$

where σ, α, c, and γ are strictly positive constants, and $A \ge 0$ is added to improve the convergence of the algorithm.

[1]For non-differentiable functions, to update the price parameters in (10), the subgradient of the objective function can be used instead of the gradient.

Algorithm 1: Price selection algorithm executed at the PCU.

1: Select initial value for σ, α, c, γ, A, and Λ^0.
2: **repeat**
3: Update σ^i and c^i as in (12).
4: **if** (FDPS) **then**
5: Calculate $\hat{g}^i(\Lambda^i)$ as in (11).
6: **elseif** (SPPS)
7: Calculate $\hat{g}^i(\Lambda^i)$ as in (13).
8: **end if**
9: Update Λ^i as in (10).
10: **until** the stopping criteria

B. Simultaneous Perturbation Price Selection (SPPS)

Next, we consider another method for approximating the gradient of the objective function $Q(\Lambda)$ which is known as *simultaneous perturbation stochastic approximation* [32, Ch. 7]. Similar to the FDPS algorithm, the SPPS algorithm updates the price parameters as in (10). However, unlike FDPS, the SPPS algorithm randomly and jointly perturbs *all* elements of Λ^i in order to obtain two different perturbed measurements of $Q(\cdot)$. Thus, the two-sided simultaneous perturbation gradient approximation is given by

$$\hat{g}^i(\Lambda^i) = \begin{bmatrix} \frac{Q(\Lambda^i + c^i\Delta^i) - Q(\Lambda^i - c^i\Delta^i)}{2c^i\Delta_1^i} \\ \vdots \\ \frac{Q(\Lambda^i + c^i\Delta^i) - Q(\Lambda^i - c^i\Delta^i)}{2c^i\Delta_p^i} \end{bmatrix} = \frac{Q(\Lambda^i + c^i\Delta^i) - Q(\Lambda^i - c^i\Delta^i)}{2c^i}\left(\frac{1}{\Delta_1^i}, \ldots, \frac{1}{\Delta_p^i}\right), \tag{13}$$

where $\Delta^i \triangleq \left(\Delta_1^i, \ldots, \Delta_p^i\right)$ is the perturbation vector, and $\Delta_j^i \in \{-1, 1\}$ is a random number. We note that, for the SPPS algorithm, the number of measurements in each iteration is two, independent of the size parameter p. Thus, compared to FDPS, the SPPS algorithm provides large savings in the number of measurements in each iteration, especially if p is large. This lower per-iteration complexity is beneficial as long as the number of iterations required to converge to an optimal value of $\Lambda^\star$ does not increase significantly.

C. Algorithm Description

In this section, we explain the steps of the proposed FDPS and SPPS algorithms (Algorithm 1) executed at the PCU. At the beginning of the algorithm, the initial values for parameters σ, α, c, γ, A, and Λ^0 are selected, c.f. Line 1. At the ith iteration of the algorithm, the coefficients σ^i and c^i are updated as in (12), c.f. Line 3. For the FDPS algorithm, the gradient is approximated as in (11), c.f. Line 5. For the SPPS algorithm, the gradient is approximated as in (13), c.f. Line 7. Λ^i is updated as in (10), c.f. Line 9. The algorithm is stopped if the maximum number of allowed iterations is reached or the difference between two subsequent values of the objective function is less than a pre-determined threshold, c.f. Line 10.

D. Convergence of the Algorithms

We now present the conditions for convergence of Algorithm 1. Convergence of different stochastic approximation based algorithms has been analyzed under various conditions. In particular, algorithms based on *simultaneous perturbation stochastic approximation* (SPSA) have attracted more attention, as they require fewer objective function evaluations. Spall showed the convergence of SPSA under a three times differentiability condition for the objective function [32]. However, it was shown later that weaker assumptions suffice for SPSA to converge [36]–[38]. The iterative updating step in (10) can be written as

$$\Lambda^{i+1} = \Lambda^i - \sigma^i\big(\tilde{g}^i(\Lambda^i) + \epsilon^i\big), \tag{14}$$

where $\tilde{g}^i(\Lambda^i)$ is any subgradient of the objective function at the ith iteration, and ϵ^i represents the observation noise and bias term. For differentiable functions, the subgradient $\tilde{g}^i(\Lambda^i)$ is identical to the gradient of the objective function. For non-differentiable functions, the sub-gradient of the objective function $Q(\cdot)$ at Λ^i is defined as any vector g that satisfies the inequality $Q(\Phi) \geq Q(\Lambda^i) + g^{\mathrm{T}}(\Phi - \Lambda^i)$ for all Φ. Following the discussion in [37], [38], the conditions for the iterative convergence of Λ^i to the optimum value Λ^* that minimizes the objective function are summarized as

- A.1 The domain of $Q(\cdot)$ is convex and closed. $Q(\cdot)$ is convex, and the expected value $\mathbb{E}[Q(\Lambda^i)]$ is uniformly bounded, where $\mathbb{E}\{\cdot\}$ denotes mathematical expectation.
- A.2 For the step-size parameters we must have: a) $\sigma^i > 0$, b) $c^i > 0$, c) $\sigma^i \to 0$, d) $c^i \to 0$, e) $\sum_{i=0}^{\infty} \sigma^i = \infty$, and f) $\sum_{i=0}^{\infty} (\sigma^i/c^i)^2 < \infty$.
- A.3 Let $\mathcal{I}^i \triangleq (\Lambda^0, \ldots, \Lambda^i)$. $\sum_{i=0}^{\infty} (\sigma^i)^2 \mathbb{E}\left[\|\epsilon^i\|^2 | \mathcal{I}^i\right] < \infty$.
- A.4 The subgradient $\tilde{g}^i$ is uniformly bounded.
- A.5 Δ_j^i must be independent for all i and j, identically distributed for all j at each iteration i, symmetrically distributed about zero, and uniformly bounded in magnitude for all i and j.

Condition A.1 specifies the criteria required for the convergence of the algorithm to the global optimum. Condition A.2 determines the rate at which the gain σ^i has to decay. The gain σ^i should decay neither too fast nor too slow. It has to approach zero fast enough to damp the effects of the noise as the algorithm gets closer to the solution Λ^*. However, it has to approach zero at a sufficiently slow rate to ensure full convergence of the algorithm. Condition A.3 ensures that the algorithm is able to cope with the noise. In practice, for large numbers of users, the effect of each individual user on the aggregate load of the system is small and the variations in the demand requirements of different users help in making the load curve smooth which also reduces the effects of the noise term. Conditions A.4 and A.5 ensure that the algorithm is asymptotically an unbiased estimator of the optimum value Λ^* [38]. Condition A.5 determines the randomization property of the perturbation vector such that the objective function can be effectively approximated by a smooth function at the points of non-differentiability [38].

Together, conditions A.1-A.5 specify the *ideal* requirements for the convergence of the algorithm. However, in practice, due to the lack of knowledge of the structure of $Q(\cdot)$, it is very difficult or even impossible to check these conditions. To resolve this issue, *gradient-free* techniques are adopted to optimize the objective function in this paper. This also reveals the difficulty of verifying the above mentioned conditions. However, despite the fact that some conditions may not be verifiable, it has been shown that the adopted techniques are among the most effective methods to optimize objective functions with an unknown formulation in practice [32]. Different methods have been proposed in the literature to ensure that the stochastic approximation methods converge to the global optimum among multiple local optima. One of the well-known approaches is to inject an additive noise in the right-hand side of the basic updating step in (10) [32, Ch. 6].

IV. System Simulator Unit (SSU)

In this section, we explain the algorithm to be implemented in the SSU. This requires an understanding of how the ECS device may operate for each user. We assume that the operation of ECS devices in each time slot begins with an *admission control* phase, where appliances send *admission requests* to the ECS unit. Once an admission request is submitted, the state of the appliance changes from *sleep* to *awake*. The ECS unit schedules the operation of awake appliances such that the electricity expenses of the user are minimized.

To simulate the users' load patterns, the SSU simulates the time at which each appliance becomes awake and also the time by which the operation of each appliance has to be finished. Such information can be obtained based on the sleep and awake history of each appliance. To preserve the users' privacy, we assume that the *actual* data is manipulated such that the *statistical information* is preserved, but it is not possible to extract the exact information about the demand requirements of individual users [39], [40]. Various privacy aware smart metering techniques have been proposed in the literature, such as secure meter data aggregation [41], and privacy aware home energy management system [42]. By using the manipulated data, the SSU simulates the likely control decisions of the ECS unit of each user based on the price indicated by the PCU. We note that the SSU simulates the likely behavior of *general users*, and each general user does not refer to any particular user.

In the following, we first explain the control algorithm running in the ECS device of each user and then how the SSU simulates the control decisions of the ECS devices.

A. Power Scheduling Done by ECS Devices

For each user u, the power scheduling is done by the ECS device at the current time slot t by solving the following optimization problem that is specific to user u and aims to minimize the *expected* energy cost in the upcoming time slots:

$$V_u^t(\mathbb{S}_u^t) = \underset{\substack{\mathbf{x}_{u,a}^t \in \mathcal{X}_u^t, \\ \forall a \in \mathcal{C}_u^k, \\ \forall k \in \mathcal{T}^t}}{\text{minimize}} \; g_t\big(\mathbb{S}_u^t, L_u^t\big) + \mathbb{E}\Big\{ V_u^{t+1}(\mathbb{S}_u^{t+1}) \mid \mathbb{S}_u^t \Big\}, \tag{15}$$

where $\mathcal{T}^t \triangleq \{t, \dots, T\}$, $\mathbf{x}_{u,a}^t \triangleq (x_{u,a}^t \dots, x_{u,a}^T)$, and we have

$$g_t\big(\mathbb{S}_u^t, L_u^t\big) \triangleq L_u^t \lambda_t\big(L_u^t\big), \tag{16}$$

$$L_u^t = \sum_{a \in \mathcal{M}_u^t} \gamma_{u,a} + \sum_{a \in \mathcal{C}_u^t} \gamma_{u,a} x_{u,a}^t. \tag{17}$$

We refer to $V_u^t(\cdot)$ as the value function of user u at time slot t, and $V_u^{T+1}(\cdot) \triangleq 0$. For each user u, we also define the *state* of the system at time slot t as $\mathbb{S}_u^t \triangleq (\mathbf{E}_u^t, \mathbf{I}_u^t)$, where $\mathbf{I}_u^t \triangleq (\mathcal{M}_u^t, \mathcal{C}_u^t)$ and $\mathbf{E}_u^t \triangleq (E_{u,1}^t, \dots, E_{u,|\mathcal{A}_u|}^t)$. Here, $\mathcal{M}_u^t$ and $\mathcal{C}_u^t$ are the sets of must-run and controllable appliances of user u that are awake at time slot t, respectively. The feasible set $\mathcal{X}_u^t$ in problem (15) is defined as

$$\mathcal{X}_u^t = \Bigg\{ \mathbf{x}_{u,a}^t \;\Bigg|\; x_{u,a}^k \in \{0,1\}, \; \forall\, a \in \mathcal{C}_u^k, \; \forall\, k \in \mathcal{T}^t, \quad \gamma_{u,a} \sum_{m=k}^{\beta_{u,a}} x_{u,a}^m = E_{u,a}^k, \; \forall\, a \in \mathcal{C}_u^k, \; \forall\, k \in \mathcal{T}^t \quad x_{u,a}^k = 1, \; \forall\, a \in \mathcal{N}_u^k, \forall\, k \in \mathcal{T}_{u,a}^t, 0 < E_{u,a}^k < E_{u,a} \Bigg\}, \tag{18}$$

where $\mathcal{T}_{u,a}^t \triangleq \{t, \dots, \beta_{u,a}\}$, and $\mathcal{N}_u^k$ denotes the non-interruptible subset of $\mathcal{C}_u^k$. The first term in the objective function in (15) is the payment of the user in the current time slot t for the *known* load L_u^t, while the second term is the expected cost of energy in the upcoming time slots, which we will refer to as the *cost-to-go*. The feasible set in (18) is similar to (4). However, it is based on the updated information which is available up to time slot t. An algorithm based on linear mixed-integer programming has been proposed in [31] to solve problem (15). However, its complexity makes it difficult to be used in the SSU.

B. Simulation of ECS Operation at SSU

In order to mimic the operation of the ECS devices, the energy provider needs to similarly solve optimization problem (15). However, this cannot be done because the energy provider does not have access to the details regarding the users' energy needs. To tackle this problem, we propose an *approximate dynamic programming* algorithm to estimate the solution of problem (15). First, we note that the state of user u in the next time slot, $\mathbb{S}_u^{t+1}$, depends on the current state $\mathbb{S}_u^t$, the decision which is made at the current time slot $\mathbf{x}_{u,a}^t$, and the exogenous information which arrives at the beginning of the next time slot $\mathbf{I}_u^{t+1}$. We define

$$\mathbb{S}_{u,x}^t = \mathrm{S}_{\mathrm{x}}(\mathbb{S}_u^t, \mathbf{x}_{u,a}^t), \tag{19}$$

$$\mathbb{S}_u^{t+1} = \mathrm{S}_{\mathrm{I}}(\mathbb{S}_{u,x}^t, \mathbf{I}_u^{t+1}), \tag{20}$$

where $\mathbb{S}_{u,x}^t$ is the state of the system immediately after we make a decision and is referred to as *post-decision* state [43], $\mathrm{S}_{\mathrm{x}}(\cdot)$ is the state transition function which takes into account the effect of decisions, and $\mathrm{S}_{\mathrm{I}}(\cdot)$ is the state transition function which takes into account the effect of arrival information.

A well-known approach to approximate the cost-to-go is to represent it based on the post-decision state $\mathbb{S}_{u,x}^t$ [43]. Problem (15) can now be written as

$$\hat{V}_u^t(\mathbb{S}_u^t) = \underset{\substack{\mathbf{x}_{u,a}^t \,\in\, \mathcal{X}_u^t,\\ \forall\, a \,\in\, \mathcal{C}_u^t}}{\text{minimize}} \;\; g_t\big(\mathbb{S}_u^t, L_u^t\big) + \hat{V}_{u,x}^{t+1}(\mathbb{S}_{u,x}^t), \tag{21}$$

where $\hat{V}_u^t(\cdot)$ is the approximation of the cost of being in state $\mathbb{S}_u^t$, and $\hat{V}_{u,x}^{t+1}(\cdot)$ is the approximation of the cost-to-go by writing it as a function of post-decision state $\mathbb{S}_{u,x}^t$ rather than current state $\mathbb{S}_u^t$. Since $\mathbb{S}_{u,x}^t$ is a deterministic function of $\mathbf{x}_{u,a}^t$, problem (21) is a deterministic optimization problem. Among different techniques considered to approximate the cost-to-go, *parametric* models [43] are particularly popular, where the value function is replaced with a *linear regression*. Let $\phi_t(\cdot)$ be a *basis function* which captures some features of the underlying system at time slot t. We approximate the cost-to-go at the next time slot as

$$\hat{V}_{u,x}^{t+1}(\mathbb{S}_{u,x}^t) = \sum_{k=t+1}^{T} \theta_k \phi_k(\tilde{\mathbf{x}}_u^{t+1}), \tag{22}$$

where θ_k is the weight coefficient at time slot k, $\tilde{\mathbf{x}}_u^{t+1} = (\tilde{\mathbf{x}}_{u,1}^{t+1}, \dots, \tilde{\mathbf{x}}_{u,|\mathcal{A}_u|}^{t+1})$, $\tilde{\mathbf{x}}_{u,a}^{t+1} = (\tilde{x}_{u,a}^{t+1}, \dots, \tilde{x}_{u,a}^{T})$, and we have

$$\phi_k(\tilde{\mathbf{x}}_u^{t+1}) = g_k(\mathbb{S}_{u,x}^t, \tilde{L}_u^k), \tag{23}$$

$$\tilde{L}_u^k = \sum_{a \in \mathcal{M}_u^t} \gamma_{u,a} + \sum_{a \in \mathcal{C}_u^t} \gamma_{u,a} \tilde{x}_{u,a}^k. \tag{24}$$

Furthermore, we can calculate $\tilde{\mathbf{x}}_u^{t+1}$ as follows:

$$\tilde{\mathbf{x}}_u^{t+1} = \underset{\mathbf{x}_{u,a}^t \,\in\, \bar{\mathcal{X}}_u^t,\, \forall\, a \,\in\, \mathcal{C}_u^t}{\text{argmin}} \;\; \sum_{k=t+1}^{T} \theta_k g_k\big(\mathbb{S}_{u,x}^t, l_u^k\big), \tag{25}$$

where $\bar{\mathcal{X}}_u^t$ is the feasible set defined by (18) while the state of the system is $\mathbb{S}_{u,x}^k$ and the first integer constraint in (18) is relaxed as $0 \le x_{u,a}^k \le 1$. l_u^k is defined as

$$l_u^k = \sum_{a \in \mathcal{M}_u^t} \gamma_{u,a} + \sum_{a \in \mathcal{C}_u^t} \gamma_{u,a} x_{u,a}^k. \tag{26}$$

The basis functions $\phi_k(\cdot)$ in (23) capture the estimate of the cost in future time slots based on the information which is available at the current time slot t. The cost-to-go then is approximated as a weighted sum of the estimated cost of all upcoming time slots. However, as the new observations about the true cost of each time slot are revealed, the weight coefficients $\boldsymbol{\theta} = (\theta_1, \dots, \theta_T)$ are updated accordingly, as we explain next.

C. Updating the Value Function Estimation

Assume that we have n different observations for the true value of being in different states (i.e., the observations from the real system at the end of the entire operation period) that can be written in vector form as $(V_u^m, \mathbb{S}_u^m)_{m=1}^n$. Let ϕ^m be the vector of basis functions evaluated at $\mathbb{S}_u^m$, and $\mathbf{\Phi}^n$ be a matrix with n rows, one corresponding to each observation, and T columns, one for each feature. Let $\mathbf{V}_u^n$ be a column vector with elements V_u^m. By using *least square* batch linear regression [43], we can update vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \big(\big(\mathbf{\Phi}^n\big)^{\mathrm{T}} \mathbf{\Phi}^n\big)^{-1} \big(\mathbf{\Phi}^n\big)^{\mathrm{T}} \mathbf{V}_u^n, \tag{27}$$

where in the above equation, T is the transpose operator. We note that at the end of the operation period, we have multiple observations for different states of the system. The estimate of the value function's parameters can be improved if the observations of multiple operating periods are used to update the $\boldsymbol{\theta}$. Moreover, the estimate of the value function's parameters can be further improved if users are able to communicate and share their observations to have more samples to update the parameters of the value function. In practice, it may not be possible to obtain the true observation of the cost-to-go from the real system because of privacy issues. To tackle this problem, the results produced by the SSU can be used to update the value function's parameters. We note that in a real system, users are making control decisions based on the partial information available at the beginning of each time slot. That is, the complete demand requirements in the future time slots are not known. The SSU simulates the behavior of each user for different scenarios. For each scenario, to better mimic the behavior of each user, the control decisions are similarly made based on partial information available at the beginning of each time slot. Thus, similar to the real system, at each time slot, the exact cost-to-go is not known and only some estimation of it is available. However, at the end of each scenario, the exact value of cost-to-go can be observed. These observations can be used instead of true observation from the real system to update the value function's parameters.

D. Algorithm Description

We now explain the steps of the proposed control algorithm (Algorithm 2) to be executed in the SSU. At the beginning, the value of n is initialized and the price parameters Λ are received from the PCU, c.f. Lines 1 and 2. Subsequently, the initial value for vector $\boldsymbol{\theta}$ is selected randomly, c.f. Line 3. For each user u and at the beginning of each time slot t, the appliances that become awake are determined. The SSU also determines the demand requirements of each appliance. That is, whether the appliance is must-run or controllable and also the deadline by which the operation of the appliance should be finished are determined. The lists of awake appliances are then updated, and the operation schedule of the awake appliances for the current time slot t is calculated as the solution of problem (21), c.f. Lines 4 to 10. At the end of the operation period, we update vector $\boldsymbol{\theta}$ as in (27). The aggregate load of the system in each time slot t is determined as $\bar{L}_t = \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}_u} \gamma_{u,a} \bar{x}_{u,a}^t$, where $\bar{x}_{u,a}^t$ is determined at time slot t as the solution of (21), c.f. Line 13.

V. Performance Evaluation

In this section, we present simulation results and assess the performance of our proposed price control algorithm. Unless stated otherwise, the simulation setting is as follows. We assume that the general RTP method combined with IBR is adopted as described in (7). We consider a system with $|\mathcal{U}| = 50$ users. Each user possesses various must-run and controllable appliances. We assume that the exact

Algorithm 2: The algorithm executed at the SSU.

Initialize n.
Receive price parameters Λ.
Select initial value for vector $\boldsymbol{\theta}$.
for $u \in \mathcal{U}$
  for $t \in \mathcal{T}$
    Determine appliances that become awake and their demand requirements.
    Receive new information $\mathbf{I}_u^t$.
    Determine $x_{u,a}^t$ as the solution of (21).
    Update $(E_{u,1}^t, \ldots, E_{u,|\mathcal{A}_u|}^t)$ as in (5).
  end for
  Update $\boldsymbol{\theta}$ as in (27).
end for
Determine the aggregate load of the system.

information about the energy requirements of the users is not known by the SSU. However, we assume that some *statistical information* about the energy requirements of the users in form of distribution functions is available at the SSU. This statistical information includes the number of appliances, the nominal power consumption of each appliance, the probabilities with which each appliance becomes awake in each time slot, and the deadline by which the operation of each appliance should be finished. The statistical information can be obtained from the operational history of the real system. In the SSU, for a typical household user, we consider on average 18 appliances. Some of the appliances and their operating specifications are summarized in Table I. The time slot at which each appliance becomes awake is selected randomly from a pre-determined interval. Based on the demand requirements of the user, each appliance can be set as must-run or controllable. This setting is decided by the user and can vary from time to time.

In our simulation setting, we consider various must-run and controllable appliances [31]. For example, we consider electric stove, clothes dryer, and vacuum cleaner as non-interruptible appliances. Refrigerator and air conditioner are modeled as interruptible appliances, and must-run appliances include: lighting, TV, etc. In general, the operation of some appliances can be correlated. However, taking such correlations into account for algorithm design would make the implementation of the SSU significantly more complex, which may not be desirable in practice. Therefore, we assume that the operations of appliances are independent. For controllable appliances, the operating deadline is selected randomly from the remaining feasible time slots.

We note that the SSU does *not* observe the demand requirements of the users in the real system. Instead, it simulates the behavior of each user by running multiple scenarios. To better simulates the decisions made by the user, for each scenario, the information about the demand requirement of the user is updated gradually over time. That is, the SSU mimics the control decisions of the user based on the partial information available at the beginning of each time slot. For each user u and at the beginning of each time slot t, we determine the appliances that become awake and their

TABLE I
OPERATING SPECIFICATIONS OF DIFFERENT APPLIANCES.

	$E_{u,a}$ (kWh)	$\gamma_{u,a}$ (kW)	arrival interval
Electric stove	4.5	1.5	[06:00, 14:00]
Clothes dryer	1	0.5	[14:00, 22:00]
Vacuum cleaner	2	1	[06:00, 15:00]
Refrigerator	2.5	0.125	[06:00, 09:00]
Air conditioner	4	1	[12:00, 22:00]
Dishwasher	2	1	[15:00, 24:00]
Heater	6	1.5	[15:00, 03:00]
Water heater	3	1.5	[06:00, 23:00]
Pool pump	4	2	[12:00, 21:00]
PEV	10	2.5	[16:00, 24:00]
Lighting	3	0.5	[16:00, 24:00]
TV	1	0.25	[16:00, 01:00]
PC	1.5	0.25	[08:00, 24:00]
Ironing appliance	2	1	[06:00, 16:00]
Hairdryer	1	1	[06:00, 13:00]
Other	6	1.5	[06:00, 24:00]

operating specifications. The lists of awake appliances are then updated, and the operation schedule of the awake appliances for the current time slot t is calculated as the solution of problem (21). The aggregate load of the system in each time slot t is determined as

$$\bar{L}_t = \sum_{u \in \mathcal{U}} \sum_{a \in \mathcal{A}_u} \gamma_{u,a} \bar{x}_{u,a}^t, \tag{28}$$

where $\bar{x}_{u,a}^t$ is obtained at time slot t as the solution of (21). This procedure is repeated for *multiple scenarios* of the demand requirements of each user and the *average* results are considered.

By testing different practical examples, it has been shown in [32] that $\alpha = 0.602$ and $\gamma = 0.101$ are good choices for (12). To mitigate the effect of the measurement noise, we set c at a level approximately equal to the standard deviation of the measurement noise in $Q(\Lambda)$. We set A equal to 10 percent of the maximum number of allowed iterations. Coefficient σ in (12) plays an important role in the convergence of the algorithm as it has a significant effect on the step size in the different iterations. To select σ, first for each element j of Λ, we determine the appropriate value of σ_j that keeps the range of changes in the jth element of Λ in an appropriate range. Second, to assure stability, we set $\sigma = \min\{\sigma_1, \ldots, \sigma_p\}$.

A. *Performance Gains for the Utility Company*

To have a baseline to compare with, we consider a system without ECS deployment, where each appliance a starts operation right after it becomes awake at its nominal power $\gamma_{u,a}$. For the system without ECS deployment, users are *not* responding to the variations of the price parameters. Furthermore, as an upper bound on the performance of the energy provider in minimizing the PAR of the aggregate load, we consider a system in which the energy provider knows all the demand requirements of the users and is capable of controlling the ECS units of all the users. The energy provider schedules the operation of all the appliances of the users such that all

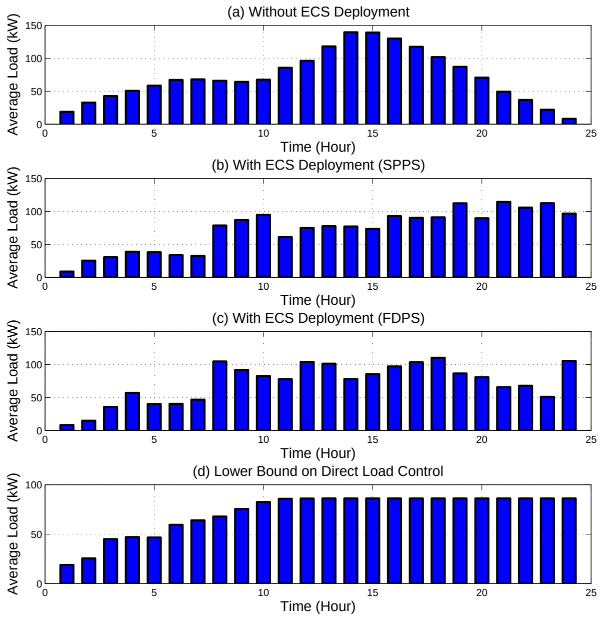


Fig. 2. Aggregate load profile in different scenarios.

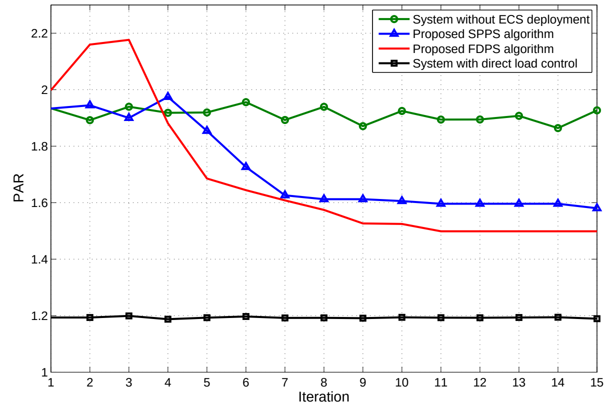


Fig. 3. The PAR of the aggregate load in different scenarios.

the demand requirements of the users are met. This system with *direct load control* achieves the minimum PAR of the aggregate load, and since the energy provider has full control over the operation of the users' appliances, the performance of the system is independent of the price parameters. We note that the existence of a pricing scheme that can achieve this performance bound is *not* guaranteed. Since optimization problem (6) is too complex, we calculate a lower bound on the PAR of the system with direct load control. That is, we treat all controllable appliances as if they are interruptible, and instead of solving the mixed integer program, we present the results for the corresponding continuous problem. Simulation results for the average total power consumption of the proposed load control algorithms, the system without ECS deployment, and the lower bound on the PAR of the system with direct load control are depicted in Fig. 2. Simulation results for the average PAR of the aggregate load at different iterations of the proposed SPPS pricing algorithm, the proposed FDPS pricing algorithm, the system without ECS deployment, and a system with direct load control are depicted in Fig. 3. The simulation results show that the PAR of the aggregate load for the system without ECS deployment is on average 1.92. Our proposed SPPS algorithm reduces the PAR of the aggregate load to 1.58 (i.e., 18% reduction). Our proposed FDPS algorithm reduces the PAR of the aggregate load to 1.49 (i.e., 22% reduction). The lower bound on the achievable PAR of the system with direct load control is on average 1.2. Considering the number of time slots and the number of price parameters for each time slot, the number of measurements of the FDPS algorithm is 72 times higher than for the SPPS algorithm.

B. Performance Gain of the Users

For the SSU, we propose a load control algorithm (Algorithm 2) which simulates the operation of the ECS unit of each user. Since the SSU has to be fast enough and has to deal with the complex system of multiple users, the proposed load control algorithm is based on an *approximate dynamic programming* approach. The proposed load control algorithm can be adopted in the ECS units of the user. Therefore, in this section, we assess the performance of the proposed load control algorithm. To have a baseline to compare with, we consider a system without ECS deployment, where each appliance a is assumed to start operation right after it becomes awake. As an upper bound, we also consider the scheme in [31] in which problem (21) is solved to schedule the operation of controllable appliances. Simulation results show that, to reduce electricity payment, the proposed control algorithm shifts the load to time slots with lower prices such as the few first hours after midnight. However, the high price penalty for exceeding the b_t threshold prevents *load synchronization* as discussed in Section I. The simulation results show that the use of the proposed algorithm reduces the average daily payment of the user from \$4.85 to \$3.99. The average daily payment of the users for the load control algorithm in [31] is \$3.88. We can see that the efficiency loss in our proposed scheme compared to the one in [31] is small, although, our design has less computational complexity and is faster. The running times of the proposed FDPS and SPPS algorithms are directly influenced by the number of measurements of the objective function in each iteration and the running time of the SSU for each measurement. The SSU simulates the load pattern of each user to produce the aggregate load pattern of the users. This process can be done in parallel or sequentially. The running time of the SSU increases approximately linearly with the number of users if the load pattern of individual users is simulated sequentially. In the following, we evaluate the complexity of the load control algorithm (Algorithm 2) which simulates the load pattern of each user for different numbers of appliances. In general, integer linear programs with n integer variables and m constraints are NP-complete. However, there exist pseudo-polynomial algorithms for solving $m \times n$ integer programs with fixed m which have a complexity of order $O(n^{2m+2}(m\nu)^{(m+1)(2m+1)}\log(n^2(m\nu^2)^{2m+3}))$, where ν is the maximum coefficient in the set of constraints [44]. A complete discussion of algorithm complexity is beyond the scope of this paper. However, to provide a general idea about the complexity of our proposed algorithm compared to the one

TABLE II
PERFORMANCE MEASURES OF DIFFERENT ALGORITHMS.

Average run time of the algorithm (in seconds).

	$\|\mathcal{A}\|$=20	$\|\mathcal{A}\|$=30	$\|\mathcal{A}\|$=40
Proposed algorithm for SSU	0.7324	0.7673	0.7919
Algorithm in [31]	2.1364	10.3071	69.5810

Average number of integer variables.

	$\|\mathcal{A}\|$=20	$\|\mathcal{A}\|$=30	$\|\mathcal{A}\|$=40
Proposed algorithm for SSU	4	6	10
Algorithm in [31]	57	90	129

in [31], simulation results for the average running time and the average number of integer variables for both algorithms are presented in Table II. The results were obtained by a computer system with Intel(R) Core(TM) i7 CPU 3.07 GHz processor, 12 GB RAM, and Windows 7 operating system.

VI. CONCLUSIONS

In this paper, we proposed two pricing algorithms based on stochastic approximation technique to minimize the PAR of the aggregate load. The proposed algorithms eliminate the need to know the structure of the objective function. In our proposed pricing algorithms, we take into account the way users will respond to different price values. We also consider the effect of control decisions of the ECS unit on the users' load profile. Moreover, we proposed the use of an SSU. A load control algorithm based on the approximate dynamic programming approach is also proposed and executed at the SSU to simulate the operation of the ECS unit at the demand side. The details of the demand requirements of the users at the appliance level are considered in the SSU. Simulation results showed that our proposed algorithms reduce the PAR of the aggregate load. The proposed algorithms provide incentives for the users to reduce their energy expenses.

In this work, we assumed that all users are equipped with ECS units and try to minimize their energy expenses. In practice, some users may schedule their power consumption to achieve different objectives such as minimizing the energy expenses, maximizing the social welfare, etc. In general, some users may be equipped with automated control units while others make control decisions manually. To obtain a better estimate of the likely behavior of the users, for the SSU, considering various users with different objectives and different levels of price-responsiveness is an interesting topic for future work.

REFERENCES

[1] C. Gellings, "The concept of demand-side management for electric utilities," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1468–1470, Oct. 1985.

[2] M. Fahrioglu and F. Alvarado, "Designing incentive compatible contracts for effective demand management," *IEEE Trans. on Power Systems*, vol. 15, no. 4, pp. 1255–1260, Nov. 2000.

[3] B. Ramanathan and V. Vittal, "A framework for evaluation of advanced direct load control with minimum disruption," *IEEE Trans. on Power Systems*, vol. 23, no. 4, pp. 1681–1688, Oct. 2008.

[4] N. Ruiz, I. Cobelo, and J. Oyarzabal, "A direct load control model for virtual power plant management," *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 959–966, May 2009.

[5] R. Faranda, A. Pievatolo, and E. Tironi, "Load shedding: A new proposal," *IEEE Trans. on Power Systems*, vol. 22, no. 4, pp. 2086–2093, Nov. 2007.

[6] A. Conejo, J. Morales, and L. Baringo, "Real-time demand response model," *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec. 2010.

[7] T. Jin and M. Mechehoul, "Ordering electricity via Internet and its potentials for smart grid systems," *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 302–310, Dec. 2010.

[8] M. Alizadeh, A. Scaglione, and R. Thomas, "From packet to power switching: Digital direct load scheduling," *IEEE J. on Selected Areas in Comm.*, vol. 30, no. 6, pp. 1027–1036, Jul. 2012.

[9] T. Logenthiran, D. Srinivasan, and T. Shun, "Demand side management in smart grid using heuristic optimization," *IEEE Trans. on Smart Grid*, vol. 3, no. 3, pp. 1244–1252, Sept. 2012.

[10] T. H. Yoo, H. G. Kwon, H. C. Lee, C. Rhee, Y. T. Yoon, and J. Park, "Development of reliability based demand response program in Korea," in *Proc. of IEEE Innovative Smart Grid Technologies*, Anaheim, CA, Jan. 2011.

[11] M. Shinwari, A. Youssef, and W. Hamouda, "A water-filing based scheduling algorithm for the smart grid," *IEEE Trans. on Smart Grid*, vol. 3, no. 2, pp. 710–719, Jun. 2012.

[12] P. Luh, Y. Ho, and R. Muralidharan, "Load adaptive pricing: An emerging tool for electric utilities," *IEEE Trans. on Automatic Control*, vol. 27, no. 2, pp. 320–329, Apr. 1982.

[13] Y. Tang, H. Song, F. Hu, and Y. Zou, "Investigation on TOU pricing principles," in *Proc. of IEEE PES Transmission and Distribution Conf. Exhibition: Asia and Pacific*, Dalian, China, Aug. 2005.

[14] M. Crew, C. Fernando, and P. Kleindorfer, "The theory of peak-load pricing: A survey," *Journal of Regulatory Economics*, vol. 8, no. 3, pp. 215–248, Nov. 1995.

[15] S. Zeng, J. Li, and Y. Ren, "Research of time-of-use electricity pricing models in China: A survey," in *Proc. of IEEE Int'l. Conf. on Industrial Engineering and Engineering Management*, Singapore, Dec. 2008.

[16] C. Chen, K. G. Nagananda, G. Xiong, S. Kishore, and L. V. Synder, "A communication-based appliance scheduling scheme for consumer -premise energy management systems," *IEEE Trans. on Smart Grid*, vol. 4, no. 1, pp. 56–65, Mar. 2013.

[17] S. Salinas, M. Li, and P. Li, "Multi-objective optimal energy consumption scheduling in smart grids," *IEEE Trans. on Smart Grid*, vol. 4, no. 1, pp. 341–348, Mar. 2013.

[18] G. Xiong, C. Chen, S. Kishore, and A. Yener, "Smart (in-home) power scheduling for demand response on the smart grid," in *Proc. of IEEE PES Innovative Smart Grid Technologies Conf.*, Anaheim, CA, Jan. 2011.

[19] R. Krishnan, "Meters of tomorrow," *IEEE Power and Energy Magazine*, pp. 92–94, Mar. 2008.

[20] A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[21] P. Samadi, R. Schober, and V. W. S. Wong, "Optimal energy consumption scheduling using mechanism design for the future smart grid," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Brussels, Belgium, Oct. 2011.

[22] P. Samadi, A. H. Mohsenian-Rad, R. Schober, V. W. S. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

[23] P. Samadi, A. H. Mohsenian-Rad, R. Schober, and V. W. S. Wong, "Advanced demand side management for the future smart grid using mechanism design," *IEEE Trans. on Smart Grid*, vol. 3, no. 3, pp. 1170–1180, Sept. 2012.

[24] J. Chen, B. Yang, and X. Guan, "Optimal demand response scheduling with Stackelberg game apprach under load uncertainty for smart grid," in *Proc. of IEEE Int'l Conf. on Smart Grid Communications*, Tainan, Taiwan, Nov. 2012.

[25] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *Proc. of IEEE Power and Energy Society General Meeting*, Detroit, MI, Jul. 2011.

[26] A. H. 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.

[27] M. Pedrasa, T. Spooner, and I. MacGill, "Scheduling of demand side resources using binary particle swarm optimization," *IEEE Trans. on Power Systems*, vol. 24, no. 3, pp. 1173–1181, Aug. 2009.

[28] S. Kishore and L. Snyder, "Control mechanisms for residential electricity demand in smartgrids," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

[29] M. Roozbehani, M. Dahleh, and S. Mitter, "Volatility of power grids under real-time pricing," *IEEE Trans. on Power Systems*, vol. 27, no. 4, pp. 1926–1940, Nov. 2012.

[30] J. Liu, Y. Xiao, S. Li, W. Liang, and C. Chen, "Cyber security and privacy issues in smart grids," *IEEE Comm. Surveys and Tutorials*, pp. 981–997, Fourth quarter 2012.

[31] P. Samadi, A. H. Mohsenian-Rad, V. W. S. Wong, and R. Schober, "Tackling the load uncertainty challenges for energy consumption scheduling in smart grid," *IEEE Trans. on Smart Grid*, vol. 4, no. 2, pp. 1007–1016, Jun. 2013.

[32] J. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, 2003.

[33] "MOSEK," *http://www.mosek.com*, 2013.

[34] P. Reiss and M. White, "Household electricity demand, revisited," *Review of Economic Studies*, vol. 72, no. 3, pp. 853–883, July 2005.

[35] S. Borenstein, "Equity effects of increasing-block electricity pricing," *Center for the Study of Energy Markets Working Paper*, vol. 180, Nov. 2008.

[36] L. Gerencser, G. Kozmann, and Z. Vago, "SPSA for non-smooth optimization with application in ECG analysis," in *Proc. of IEEE Conference on Decision and Control*, Tampa, FL, Dec. 1998.

[37] Y. He, M. C. Fu, and S. I. Marcus, "Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization," *IEEE Trans. on Automatic Control*, vol. 48, no. 8, pp. 1459–1463, Aug. 2003.

[38] F. Yousefian, A. Nedić, and U. V. Shanbhag, "On stochastic gradient and subgradient methods with adaptive steplength sequences," *Automatica*, vol. 48, no. 1, pp. 56–67, Jan. 2012.

[39] Y. Kim, E. Ngai, and M. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Brussels, Belgium, Oct. 2011.

[40] T. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, "PASS: Privacy-preserving authentication scheme for smart grid network," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Brussels, Belgium, Oct. 2011.

[41] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

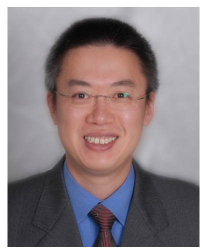
[42] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Proc. of IEEE Int'l. Conf. on Smart Grid Communications*, Gaithersburg, MD, Oct. 2010.

[43] W. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Blackwell, 2007.

[44] C. Papadimitriou, "On the complexity of integer programming," *J. of the ACM*, vol. 28, no. 4, pp. 765–768, Oct. 1981.

Pedram Samadi (S'09) received the B.Sc. and the M.Sc. degrees both from Isfahan University of Technology, Isfahan, Iran in 2006 and 2009, respectively. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. His research interests are in the area of smart grid and especially demand side management.

Hamed Mohsenian-Rad (S'04, M'09) received masters degree in Electrical Engineering from Sharif University of Technology in 2004 and Ph.D. degree in Electrical and Computer Engineering from The University of British Columbia in 2008. Currently, he is an Assistant Professor of Electrical Engineering at the University of California at Riverside. Dr. Mohsenian-Rad is the recipient of the NSF CAREER Award, NSERC Post-doctoral Fellowship, and the Best Paper Award from the *IEEE International Conference on Smart Grid Communications 2012*. He is an Associate Editor of the *IEEE Communications Surveys and Tutorials*, a Guest Editor of the *ACM Transactions on Embedded Computing Systems* - Special Issue on Smart Grid, and a Guest Editor of the *KICS/IEEE Journal of Communications and Networks* - Special Issue of Smart Grid. His research interests include the design, optimization, and game-theoretic analysis of power systems and smart grid.



Vincent W.S. Wong (SM'07) 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. 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 the Internet, wireless networks, smart grid, RFID systems, and intelligent transportation systems. Dr. Wong is an Associate Editor of the *IEEE Transactions on Communications* and *IEEE Transactions on Vehicular Technology*. He has served as an Editor of *Journal of Communications and Networks*. Dr. Wong is the Symposium Co-chair of *IEEE SmartGridComm'13 – Communications Networks for Smart Grid and Smart Metering Symposium*, and *IEEE Globecom'13 – Communication Software, Services, and Multimedia Application Symposium*.

Robert Schober (S'98, M'01, SM'08, F'10) was born in Neuendettelsau, Germany, in 1971. He received the Diplom (Univ.) and the Ph.D. degrees in electrical engineering from the University of Erlangen-Nuermberg in 1997 and 2000, respectively. From May 2001 to April 2002 he was a Postdoctoral Fellow at the University of Toronto, Canada, sponsored by the German Academic Exchange Service (DAAD). Since May 2002 he has been with the University of British Columbia (UBC), Vancouver, Canada, where he is now a Full Professor and Canada Research Chair (Tier II) in Wireless Communications. Since January 2012 he is an Alexander von Humboldt Professor and the Chair for Digital Communication at the Friedrich Alexander University (FAU), Erlangen, Germany. His research interests fall into the broad areas of Communication Theory, Wireless Communications, and Statistical Signal Processing.

Dr. Schober received the 2002 Heinz MaierLeibnitz Award of the German Science Foundation (DFG), the 2004 Innovations Award of the Vodafone Foundation for Research in Mobile Communications, the 2006 UBC Killam Research Prize, the 2007 Wilhelm Friedrich Bessel Research Award of the Alexander von Humboldt Foundation, the 2008 Charles McDowell Award for Excellence in Research from UBC, a 2011 Alexander von Humboldt Professorship, and a 2012 NSERC E.W.R. Steacie Fellowship. In addition, he received best paper awards from the German Information Technology Society (ITG), the European Association for Signal, Speech and Image Processing (EURASIP), *IEEE WCNC 2012*, *IEEE Globecom 2011*, *IEEE ICUWB 2006*, the *International Zurich Seminar on Broadband Communications*, and *European Wireless 2000*. Dr. Schober is a Fellow of the Canadian Academy of Engineering and a Fellow of the Engineering Institute of Canada. He is currently the Editor-in-Chief of the *IEEE Transactions on Communications*.