Dynamic Pricing and Energy Consumption Scheduling with Reinforcement Learning

Byung-Gook Kim*, Yu Zhang†, Mihaela van der Schaar†, and Jang-Won Lee*
*Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
†Department of Electrical Engineering, UCLA, Los Angeles, USA

***Abstract*—In this paper, we study a dynamic pricing and energy consumption scheduling problem in the microgrid where the service provider acts as a broker between the utility company and customers by purchasing electric energy from the utility company and selling it to the customers. For the service provider, even though dynamic pricing is an efficient tool to manage the microgrid, the implementation of dynamic pricing is highly challenging due to the lack of the customer-side information and the various types of uncertainties in the microgrid. Similarly, the customers also face challenges in scheduling their energy consumption due to the uncertainty of the retail electricity price. In order to overcome the challenges of implementing dynamic pricing and energy consumption scheduling, we develop reinforcement learning algorithms that allow each of the service provider and the customers to learn its strategy without a priori information about the microgrid. Through numerical results, we show that the proposed reinforcement learning-based dynamic pricing algorithm can effectively work without a priori information about the system dynamics and the proposed energy consumption scheduling algorithm further reduces the system cost thanks to the learning capability of each customer.**

***Index Terms*—Smart grid, microgrid, dynamic pricing, load scheduling, demand response, electricity market, Markov decision process, reinforcement learning.**

NOMENCLATURE

Sets and Parameters:

γ	Discount factor of Markov decision process
λ_i	Demand backlog rate of customer i
$\mathcal{A}$	Set of actions of service provider (retail pricing functions)
$\mathcal{A}_i$	Set of actions of customer i (energy consumption scheduling)
$\mathcal{C}$	Set of cost functions of service provider
$\mathcal{D}_i$	Set load demand levels of customer i
$\mathcal{H}$	Set of periods
$\mathcal{I}$	Set of customers
$\mathcal{S}$	Set of system states of microgrid
$\mathcal{S}_i$	Set of states of customer i
$\mathcal{X}$	Set of energy consumption-based approximate states (EAS)
μ_i	Demand extension for rate of customer i
ρ	Weighting factor between costs of service provider and customers
e_i^{max}	Maximum amount of energy consumption which can be consumed by customer i during each time-slot
p_c	Transition probability of cost function
p_s	Transition probability of state
p_x	Transition probability of EAS
p_{d_i}	Transition probability of load demand
$p_{d_{app}}$	Transition probability of approximate demand

Variables:

ϕ_i^t	Cost of customer i at time-slot t
π	Stationary policy of service provider
π_i	Stationary policy of customer i
ψ^t	Cost of service provider at time-slot t
σ^t	Experience tuple at time-slot t
θ	Set of virtual experience tuples
$\tilde{\sigma}^t$	Virtual experience tuple at time-slot t
a^t	Retail price function at time-slot t
c^t	System cost function at time-slot t
D_i^t	New load demand of customer i at time-slot t
d_i^t	Accumulated load demand of customer i at time-slot t
d_{app}^t	Approximate demand at time-slot t
e_i^t	Customer i's energy consumption decision at time-slot t
h^t	Period at time-slot t
r^t	System cost at time-slot t
s^t	System state of microgrid at time-slot t
s_i^t	State of customer i at time-slot t
t	Index of time-slot
u_i	Disutility function of customer i
x^t	EAS at time-slot t

The earlier version of this paper was presented at IEEE CCSES 2014[1].
This work was supported in part by Mid-career Researcher Program through NRF grant funded by the MSIP, Korea (2013R1A2A2A01069053).

I. Introduction

In the smart grid system, thanks to the real-time information exchange through communication networks, customers can schedule the operation of their appliances according to the change of electricity price via the automated energy management system equipped in households, which we refer to as *demand response* [2].

A natural realization of demand response is the energy consumption scheduling of residential appliances which is one of the most actively studied research topics in smart grid. From the customers' perspective, the majority of the previous works on the energy consumption scheduling focus on directly controlling the energy consumption of the residential appliances to maximize the social welfare of the smart grid system under a given pricing policy. For example, in [3][4], the energy consumption scheduling of various types of appliances was studied considering Time Of Use (TOU) pricing. In [5], reinforcement learning algorithm was adoped to cope with the randomness of temperature under the fixed TOU pricing. In [6] and [7], utility maximization problems were studied by using auction and non-cooperative game approaches where the price is determined by the interactions among the customers. Many recent works [8][9][10][11][12][13][14] considered Real Time Pricing (RTP) and introduced various types of price prediction schemes such as filtering-based price prediction [8], opportunistic energy scheduling [9], robust optimization [10][11], and reinforcement learning [12][13][14]. However, the common assumption in these works is that pricing policies deployed by the service providers are predetermined. For example, in [3][4][5][6][7], it is assumed that the price functions are fixed during a certain period. In [8], although real-time pricing was considered, the authors only added the restriction that the upcoming prices are not announced to the customer in advance under the assumption that the pricing functions are predetermined and fixed. In [9][10][11][12][13][14], the pricing functions are assumed to follow a certain random process.

On the contrary, in this paper, we consider that the service provider decides what dynamic pricing policies to adopt to enable more efficient energy consumption. We first consider a scenario where the service provider can adaptively decide the retail electricity price based on the customers' load demand level and the cost of electricity from the utility company to minimize either the customers' disutility (in the case of a benevolent service provider) or its own cost (in the case of a profit-making service provider).

Recently, there have been several works on dynamic pricing for smart grid [15][16][17][18][19][20][21][22][23]. In [15] and [16], dynamic pricing problems were studied aiming at maximizing the social welfare considering a smart grid system with multiple residences and a single service provider. In [17], the authors developed an incentive-based dynamic pricing scheme which allows the service provider to decide the incentive for the customers who shift their appliances' usage from peak hours to off-peak hours. The authors in [18] focused on the smart grid system with non-cooperative customers where the conventional optimization approach cannot be applied and developed a simulated annealing-based dynamic pricing algorithm. In a similar context, in [19] the dynamic pricing problem was modeled as a Stackelberg game where the service provider decides the retail price and each selfish customer decides the schedule for its appliances according to the price. Recently, the game theoretic approach to the dynamic pricing was extended to a variety of types such as the multi-stage game for the time-slotted system in [20], the auction game between the service provider and customers in [21], and the two-level game with multiple utility companies in [22]. In [23], the authors took into account the uncertainties of energy supply and demand while formulating an Markov decision process (MDP) problem and developed an online algorithm.

Despite those previous efforts, there still exist several critical challenges in implementing dynamic pricing for demand response. First, in the practical smart grid system, it is not easy for the service provider to obtain the customer-side information such as their current load demand levels and the transition probability of the demand levels, and the customer-specific utility models including the willingness to purchase electric energy given their load demand level and retail price. Second, even if the service provider can obtain those information, the service provider which lies between the utility company and the customers may not obtain the perfect information about the amount of actual energy that the customers will consume. Finally, it is challenging for the service provider to have the ability to estimate the impact of its current pricing decision on the customers' future behavior. Consequently, most of existing works on dynamic pricing for smart grid have been studied in myopic approaches where the algorithms for dynamic pricing and demand side energy consumption scheduling are conducted within a given time period without considering the long-term performance of the smart grid system.

In order to overcome the aforementioned challenges of dynamic pricing, in this paper, we use reinforcement learning to allow the service provider to learn the behaviors of customers and the change of electricity cost to make an optimal pricing decision [1]. We consider various stochastic dynamics of the smart grid system including the customers' dynamic demand generation and energy consumptions, and changes of electricity cost from the utility company. Considering system model for microgrid, we formulate an MDP problem where the service provider observes the system state transition and decides the retail electricity price to minimize its expected total cost or the customers' disutility. To solve the MDP problem, we adopt the Q-learning algorithm with proposing two improvements: alternative state definition and virtual experience.

We then extend our system model to a more intelligent microgrid system by adopting multi-agent learning structure where each customer can decide its energy consumption scheduling based on the observed retail price aiming at min-

[1]Reinforcement learning in smart grid was studied in [12], but it is very different from this paper as the study in [12] is focused on the demand-side storage management in a single household while this paper addresses the dynamic pricing of the service provider as well as the energy consumption scheduling of a number of customers which are associated with both electricity cost from various energy sources and retail electricity markets.

TABLE I
COMPARISON WITH RELATED WORKS ON DYNAMIC PRICING.($\surd$: CONSIDERED, -:NOT CONSIDERED)

	Demand uncertainty	Uncertainty of electricity cost	Dynamic pricing without explicit customer-side information	Learning capability on energy consumption scheduling
[20]	-	-	-	-
[15],[19],[21],[22]	-	-	$\surd$	-
[16]	$\surd$	-	-	-
[17],[18],[23]	$\surd$	-	$\surd$	-
Our work	$\surd$	$\surd$	$\surd$	$\surd$

imizing its expected cost. To the best of our knowledge, this is the first paper that investigates the multi-agent learning in the smart grid system including both the service provider's dynamic pricing and the customers energy consumption scheduling. While some previous works [24][25][26] have considered a smart grid system with multiple customers trying to predict the retail price, they assumed that the probability distribution of the retail price is known to the customers. Moreover, they considered only a fixed pricing rule and did not considered the learning problem of the service provider. In contrast, in this paper, we consider the learning problem of both the service provider and the customers: the service provider aims to learn to minimize the system cost while the customers aim to individually minimize their own costs. We develop a reinforcement learning-based energy consumption scheduling algorithm which can be conducted in a fully distributed manner at each customer along with the proposed dynamic pricing algorithm for the service provider. In order to enhance each customer's learning speed, we adopt a post decision state (PDS) learning algorithm.

We summarize the comparison of our work with the existing works focusing on dynamic pricing for the smart grid system in Table I.

The rest of this paper is organized as follows. In Section II, the system model is presented. In Section III, we define a dynamic pricing problem and develop a reinforcement learning-based dynamic pricing algorithm. In Section IV, we develop a reinforcement learning-based energy consumption scheduling algorithm by modeling the learning capability of customers. We provide numerical results in Section V and finally conclude in Section VI.

II. SYSTEM MODEL

We consider a microgrid system which consists of one service provider, a set of customers $\mathcal{I}$ as in Fig. 1. The microgrid operates in a time-slotted fashion, where each time-slot has an equal duration. At each time-slot t, the service provider buys electric energy from the utility company and provides it to the customers through a retail electricity market. In the retail electricity market, at each time-slot t, the service provider determines the retail pricing function $a^t : \mathbb{R}_+ \to \mathbb{R}_+$ and charges each customer i an electricity bill $a^t(e_i^t)$, where e_i^t denotes customer i's energy consumption at time-slot t. We define the set of retail pricing functions as $\mathcal{A} = \{a_1, a_2, \cdots, a_A\}$ and assume that the number of retail pricing functions, A, is finite.

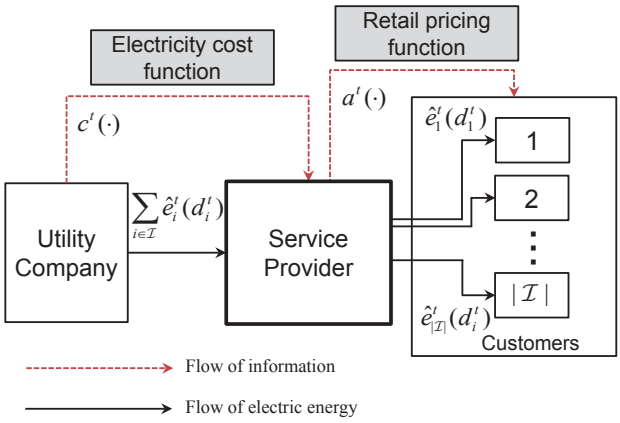

Fig. 1. Microgrid system.

We assume that the customers' average demand generation rate and the cost function of the service provider at a time-slot can vary depending on its actual time in a day. To model this time-dependency, we introduce a set of periods $\mathcal{H} = \{0, 1, \cdots, H-1\}$ each of which represents an actual time in a day. We map each time-slot t to one period $h \in \mathcal{H}$ denoting the period at time-slot t by h^t. We assume that the sequence of periods $h^t, t = 0, 1, 2, \cdots$ is predetermined in a deterministic manner and repeated every day, i.e.,

$$h^t = \mod(t, H), \quad \forall t \geq 0. \tag{1}$$

A. Model of Customer's Response

In each time-slot, each customer has an *accumulated* load demand [2], which is defined as the total amount of energy that it wants to consume for its appliances in that time-slot. We denote the amount of accumulated load demand of customer i at time-slot t by $d_i^t \in \mathcal{D}_i$, where $\mathcal{D}_i$ is the set of customer i's accumulated load demand levels. Once customer i consumes energy e_i^t at time-slot t, the corresponding amount of customer i's load demand is satisfied and the rest of the accumulated load demand $d_i^t - e_i^t$ is not satisfied which we call *remaining* load demand. The remaining load demand causes dissatisfaction to the customer at that time-slot, which is denoted by a disutility function for $u_i : \mathbb{R}_+ \to \mathbb{R}_+$. We assume that $u_i(\cdot)$ is an increasing convex function of the remaining load demand $d_i^t - e_i^t$. For the purpose of service management, each customer's disutility can be reported to the service provider at each time-slot.

[2]For the convenience, 'accumulated load demand' and 'load demand' are used interchangeably in the rest of this paper.

Based on the disutility $u_i(d_i^t - e_i^t)$ and the electricity bill $a^t(e_i^t)$, we define customer i's cost at each time-slot t as

$$\phi_i^t(d_i^t, e_i^t) = u_i(d_i^t - e_i^t) + a^t(e_i^t). \quad (2)$$

We assume that a portion of each customer's remaining load demand at a time-slot is carried forward to the next time-slot. This process is modeled by adopting the concept of *demand backlog* which is represented as $\lambda_i(d_i^t - e_i^t)$, where $0 \leq \lambda_i \leq 1$ is the backlog rate of load demand. Similarly, in order to consider other situations such as some appliances that operated in the previous time slot still want to operate in the current time slot, we adopt the concept of *demand extension* which is represented as $\mu_i e_i^t$, where $0 \leq \mu_i \leq 1$ is the extension rate of load demand. At each time-slot t, each customer i randomly generates its *new* load demand, D_i^t, and its distribution is assumed to be dependent on the current period h^t. Note that our reinforcement learning algorithms which will be developed in Sections III and IV are able to operate efficiently regardless of the types of the demand arrival model. At the beginning of each time-slot $t+1$, customer i's accumulated load demand d_i^{t+1} is updated as

$$d_i^{t+1} = \lambda_i(d_i^t - e_i^t) + \mu_i e_i^t + D_i^{t+1}. \quad (3)$$

We denote the transition probability of load demand by $p_{d_i}(d_i^{t+1}|d_i^t, h^t, a^t)$

B. Electricity Cost of Service Provider

At each time-slot t, the service provider buys electric energy, which corresponds to the total amount of energy consumption of customers, $\sum_{i\in\mathcal{I}} e_i^t$, from the utility company as illustrated in Fig. 1. The cost charged to the service provider is determined based on a cost function $c^t : \mathbb{R}_+ \to \mathbb{R}_+$, where c^t is a function of the total amount of energy consumption $\sum_{i\in\mathcal{I}} e_i^t$. We assume that c^t is selected among a finite number of cost functions in set $\mathcal{C}$ and its transition probability from c^t to c^{t+1} depends on the current cost function, c^t, and the current period, h^t, and thus it can be represented as $p_c(c^{t+1}|c^t, h^t)$. In this paper, we adopt a simplified cost model of the service provider by choosing a single and random cost function rather than fully modeling the cost model for electricity from the utility company. However, we do not impose any assumption on the shape of the cost function, c^t.

We define the service provider's cost at each time-slot t as a function of the customers' load demand vector $\bar{d}^t = [d_i^t]_{i\in\mathcal{I}}$, the electricity cost function c^t, and the retail pricing function a^t, i.e.,

$$\psi^t(\bar{d}^t, c^t, a^t) = c^t\Big(\sum_{i\in\mathcal{I}} e_i^t\Big) - \sum_{i\in\mathcal{I}} a^t(e_i^t), \quad (4)$$

where the first term denotes the total electricity cost of the service provider and the second term denotes the service provider's revenue from selling energy to the customers.

In Fig. 2, we illustrate the timeline of the interaction among the microgrid components including the service provider's decision on the retail pricing function, the customers' response, and the change of the cost function of the microgrid.

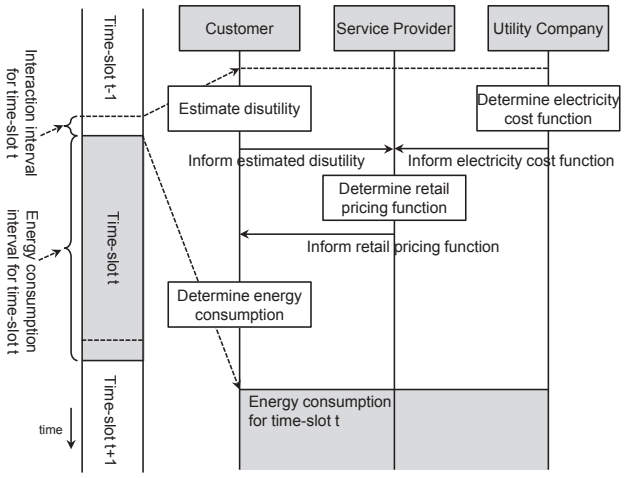

Fig. 2. Timeline of interaction among the microgrid components.

III. Reinforcement Learning at Service Provider: Dynamic Pricing Algorithm

In this section, based on the microgrid system introduced in the previous section, we first formulate a dynamic pricing problem in the framework of MDP. Then, by using reinforcement learning, we develop an efficient and fast dynamic pricing algorithm which does not require the information about the system dynamics and uncertainties.

A. Problem Formulation

We formulate the dynamic pricing problem in the microgrid system as an MDP problem, which is defined by a set of decision maker's actions, a set of system states and their transition probabilities, and a system cost function for the decision maker. In our MDP problem, the decision maker is the service provider whose action is choosing a retail pricing function $a^t \in \mathcal{A}$ at each time-slot t. In this section, we focus on the decision making of the service provider and assume that the customers are myopic and deterministic, i.e., each customer tries to minimize its cost at each time-slot. Then, we represent customer i's energy consumption decision that minimizes its cost [3] as

$$e_i^t = \underset{0\leq e\leq \min(e_i^{max}, d_i^t)}{\operatorname{argmin}} \phi_i^t(d_i^t, e), \quad (5)$$

where e_i^{max} is the maximum amount of energy that can be consumed at each time-slot which is determined by physical limitations of the microgrid. Note that we will extend this model to consider the case in which each customer is able to learn to minimize its expected long-term cost in Section IV.

We define the system state of the microgrid at time-slot t as a combination of the accumulated load demands vector, $\bar{d}^t$, the current period h^t, and the cost function, c^t, i.e.,

$$s^t = (\bar{d}^t, h^t, c^t) \in \mathcal{S}, \quad (6)$$

[3]Customer i's energy consumption decision is a function of its current load demand d_i^t and the retail pricing function a^t. However, for the simple expression, we represent it as e_i^t without indicating d_i^t and a^t.

where $\mathcal{S} = \prod_{i\in\mathcal{I}} \mathcal{D}_i \times \mathcal{H} \times \mathcal{C}$. Since the transition of each customer's load demand, that of the period, and that of the cost function depend only on the state s^t and action a^t at time-slot t, the sequence of states $\{s^t, t = 0, 1, 2, \cdots\}$ follows a Markov decision process with action a^t. The transition probability from state $s^t = (\bar{d}^t, h^t, c^t)$ to state $s^{t+1} = (\bar{d}^{t+1}, h^{t+1}, c^{t+1})$ with given action a^t can be represented as

$$p_s(s^{t+1}|s^t, a^t) = p_c(c^{t+1}|c^t, h^t) \times \prod_{i\in\mathcal{I}} p_{d_i}(d_i^{t+1}|d_i^t, h^t, a^t).$$

We define the system cost for the service provider at each time-slot t as the weighted sum of the service provider's cost and the customers' cost at the time-slot:

$$r^t(s^t, a^t) = (1-\rho)\psi^t(\bar{d}^t, c^t, a^t) + \rho \sum_{i\in\mathcal{I}} \phi_i^t(d_i^t, e_i^t), \quad (7)$$

where $\rho \in [0, 1]$ denotes the weighting factor that determines the relative importance between the service provider's cost and the customers' cost. That is, with a larger ρ, the service provider puts more importance on minimizing the customers' cost whereas, with a smaller ρ, the service provider puts more importance on minimizing the its own cost.

We denote the stationary policy that maps states to actions (retail pricing functions) by $\pi : \mathcal{S} \to \mathcal{A}$, i.e, $a^t = \pi(s^t)$. The objective of our dynamic pricing problem is to find an optimal policy π^* for each state $s \in \mathcal{S}$ that minimizes the expected discounted system cost of the service provider as in the following MDP problem $\mathbf{P}$:

$$\mathbf{P} : \min_{\pi:\mathcal{S}\to\mathcal{A}} E\left[\sum_{t=0}^{\infty} (\gamma)^t r^t(s^t, \pi(s^t))\right], \quad (8)$$

where $0 \le \gamma < 1$ is the discount factor which represents the relative importance of the future system cost compared with the present system cost.

The optimal stationary policy π^* can be well defined by using the optimal *action-value function* $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which satisfies the following Bellman optimality equation:

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s, a)V^*(s'), \quad (9)$$

where $V^*(s')$ is the optimal *state-value function* [27], which is defined as

$$V^*(s') = \min_{a\in\mathcal{A}} Q^*(s', a), \forall s \in \mathcal{S}. \quad (10)$$

Since $Q^*(s, a)$ is the expected discounted system cost with action a in state s, we can obtain the optimal stationary policy as

$$\pi^*(s) = \operatorname*{argmin}_{a\in\mathcal{A}} Q^*(s, a). \quad (11)$$

In solving our MDP problem $\mathbf{P}$, we use the well-known Q-learning algorithm as a baseline, by which we can solve $\mathbf{P}$ without acquisition of the state transition probabilities $p_s(s^{t+1}|s^t, a^t), \forall s \in \mathcal{S}$ a priori. We refer the readers to [28] for more detail on the Q-learning algorithm. In the subsequent subsections, we introduce the existing drawbacks of the conventional Q-learning algorithm in practical microgrid system and propose two improvements that resolve them.

B. Energy Consumption-Based Approximate State (EAS)

There are two obstacles in implementing the Q-learning algorithm in the microgrid system. First, the number of system states is very large which makes the Q-learning algorithm require not only a large memory space to store the state-action function $Q(s, a)$, but also a long time for the convergence. Second, in the practical microgrid system, it is difficult for the service provider to acquire or use the information about the customers' current load demands due to the privacy issue. In order to resolve these difficulties, in this section, we propose an alternative definition of the system state, which is based on the observed total energy consumption and the previously chosen retail price. For notational convenience, we will omit d_i^t in e_i^t in the rest of this subsection.

The main idea of this alternative state definition comes from the fact that each customer i's energy consumption at time-slot $t-1$, e_i^{t-1}, is determined by its accumulated load demand d_i^{t-1} and the retail pricing function a^{t-1} as in (2) and (5). Hence, given a^{t-1}, a different energy consumption e^{t-1} implies a different load demand at that time-slot. Now, by the load demand update process in (3), if the new load demand D_i^t at the current time-slot is known, a different energy consumption e^{t-1} implies a different current load demand d_i^t. Similarly, once a tuple $(\sum_{i\in\mathcal{I}} e_i^{t-1}, a^{t-1}, \sum_{i\in\mathcal{I}} D_i^t)$ is observed by the service provider, it approximately reflects the customers' overall load demands at time-slot t. Since D_i^t is independent random variable for each customer i, by the law of the large number, the normalized value of sum of new load demand, $\sum_{i\in\mathcal{I}} D_i^t/|\mathcal{I}|$, goes to its expected value as the number of customers gets larger. This implies that in the practical microgrid system with a large number of customers, a tuple $(\sum_{i\in\mathcal{I}} e_i^{t-1}, a^{t-1})$ provides enough information for the service provider to infer the customers' overall load demand level at time-slot t.

To reduce the number of system states, we discretize the observed energy consumption $\sum_{i\in\mathcal{I}} e_i^{t-1}$ into a finite number of energy levels in $\mathcal{E}$ by using a quantization operation $q_{\mathcal{E}}(\cdot)$. Then, we refer to tuple $(q_{\mathcal{E}}(\sum_{i\in\mathcal{I}} e_i^{t-1}), a^{t-1})$ as the *approximate demand* at time-slot t and represent it as

$$d_{app}^t = \Big(q_{\mathcal{E}}\Big(\sum_{i\in\mathcal{I}} e_i^{t-1}\Big), a^{t-1}\Big). \quad (12)$$

Based on the approximate demand, we now define the *energy consumption-based approximate state (EAS)* of the microgrid as

$$x^t = (d_{app}^t, h^t, c^t) \in \mathcal{X}, \quad (13)$$

where $\mathcal{X} = \mathcal{E} \times \mathcal{A} \times \mathcal{H} \times \mathcal{C}$ denotes the set of the EASs. Then, the transition probability of the EAS from x^t to x^{t+1} can be obtained as

$$p_x(x^{t+1}|x^t, a^t) = p_c(c^{t+1}|c^t)p_{d_{app}}(d_{app}^{t+1}|d_{app}^t, a^t), \quad (14)$$

where $p_{d_{app}}(d_{app}^{t+1}|d_{app}^t, a^t)$ is the transition probability of the approximate demand. Note that the EAS extremely reduces the number of states from $|\mathcal{S}| = |\prod_{i\in\mathcal{I}} \mathcal{D}_i \times \mathcal{H} \times \mathcal{C}|$ to $|\mathcal{X}| = |\mathcal{E} \times \mathcal{A} \times \mathcal{H} \times \mathcal{C}|$, while allowing the service provider to easily infer the customers' current load demand level without using direct signaling from the customers. Now, we can simply substitute the original state s^t by EAS x^t in the Q-learning algorithm.

Algorithm 1 Q-Learning Algorithm with Virtual Experience

1: Initialize Q arbitrarily, $t = 0$
2: **for each time-slot** t
3: Choose a^t according to policy $\pi(x^t)$
4: Take action a^t, observe system cost $r(x^t, a^t)$ and next state x^{t+1}
5: Obtain experience tuple $\sigma^{t+1} = (x^t, a^t, r^t, x^{t+1})$
6: Generate set of virtual experience tuples $\theta(\sigma^{t+1})$
7: **for each virtual experience tuple** $\tilde{\sigma}^{t+1} \in \theta(\sigma^{t+1})$
8: $v = r(x^t, a^t) + \gamma \max_{a' \in \mathcal{A}} Q(x^{t+1}, a')$
9: $Q(x^t, a^t) \leftarrow (1 - \epsilon) Q(x^t, a^t) + \epsilon v$
10: **end**
11: **end**

C. Accelerated Learning using Virtual Experience

In order to improve the speed of the Q-learning algorithm, we adopt virtual experience update which was introduced in [29]. The virtual experience update enables the service provider to update multiple state-action pairs at each time-slot by exploiting a priori known partial information about the state transition probability. In this subsection, we consider the case where the service provider knows the transition probability of the cost function $p_c(c^{t+1}|c^t, h^t)$ a priori [4].

We first define the *experience tuple* observed by the service provider at time-slot $t+1$ as $\sigma^{t+1} = (x^t, a^t, r^t, x^{t+1})$, where r^t is the observed system cost. While the observed experience tuple σ^{t+1} is used to update only one state-action function $Q(x^t, a^t)$ in the conventional Q-learning, we can generate multiple *virtual experience tuples*, which are statistically equivalent to the actual experience tuple, to update multiple state-action functions simultaneously. An experience tuple $\tilde{\sigma}^{t+1} = (\tilde{x}^t, \tilde{a}^t, \tilde{r}^t, \tilde{x}^{t+1})$ is said to be statistically equivalent to another experience tuple $\sigma^{t+1} = (x^t, a^t, r^t, x^{t+1})$ if $p_x(\tilde{x}^{t+1}|\tilde{x}^t, \tilde{a}^t) = p_x(x^{t+1}|x^t, a^t)$, $\tilde{a}^t = a^t$, and the system cost $\tilde{r}^t$ can be calculated by using $\tilde{\sigma}^{t+1}$.

We denote a virtual experience tuple by $\tilde{\sigma}^{t+1} = (\tilde{x}^t, \tilde{a}^t, \tilde{r}^t, \tilde{x}^{t+1}) \in \theta(\sigma^{t+1})$, where $\theta(\sigma^{t+1})$ represents the set of virtual experience tuples which are statistically equivalent to the actual experience tuple σ^{t+1}. Note that, given approximate demand d^t_{app}, period h^t, and retail electricity pricing function a^t, the customers' total energy consumption $\sum_{i \in \mathcal{I}} e^t_i$ is the same regardless of the cost function c^t. Hence, if d^t_{app} and h^t are fixed, the system cost r^t can be easily calculated for an arbitrary cost functions $c'^t \in \mathcal{C}$ by applying the same energy consumption $\sum_{i \in \mathcal{I}} e^t_i$ to (7). We represent the virtually calculated system cost as $r(c'^t)$ where c'^t is an arbitrary cost function. Then, given the actual experience tuple $\sigma^{t+1} = (x^t, a^t, r^t, x^{t+1})$, we can obtain the set of the corresponding virtual experience tuples [5] as

$$\theta(\sigma^{t+1}) = \left\{ \tilde{\sigma}^{t+1} \;\middle|\; \begin{array}{l} \tilde{d}^t_{app} = d^t_{app}, \tilde{h}^t = h^t, \\ \tilde{a}^t = a^t, \tilde{r}^t = r(\tilde{c}^t), \\ p_c(\tilde{c}^{t+1}|\tilde{c}^t, \tilde{h}^t) = p_c(c^{t+1}|c^t, h^t) \end{array} \right\}. \quad (15)$$

For example, the coarser the discrete price set is, the more likely there are a large number of virtual experience tuples in $\theta(\sigma^{t+1})$. Moreover, if the price variation across time-slots is small, the number of virtual experience tuples will become even larger. Hence, as long as the number of discrete prices and the price variation is limited, $\theta(\sigma^{t+1})$ will have sufficient tuples to take advantage of the virtual experience update.

By using the virtual experience tuples, the Q-learning algorithm can update multiple state-action pairs at each time-slot as outlined in Algorithm 1. Lines 3-5 describe the operation of the conventional Q-learning where the service provider takes an action and evaluates the corresponding system cost and state transition, i.e., obtain the experience tuple σ^{t+1}. Then, in line 6, based on σ^{t+1}, a set of its virtual experiences is generated. In lines 7-10, the action-value function $Q(x^t, a^t)$ is updated for all virtual experience tuples in $\theta(\sigma^{t+1})$.

The computational complexity and the memory complexity of the proposed reinforcement learning algorithms are summarized in Table II. The computational complexity of the Q-learning algorithm with virtual experience at each time-slot is $O(|\theta(\sigma^{t+1})||\mathcal{A}|)$, where $|\theta(\sigma^{t+1})|$ is determined by the transition probability of the cost function of the service provider and the current actual experience tuple σ^{t+1}. Although the Q-learning algorithm with virtual experience has a higher update complexity than the conventional Q-learning algorithm, it significantly reduces the number of time-slots needed to converge, which is, in general, regarded as a more important aspect than the computational complexity at each time-slot in reinforcement learning algorithms.

TABLE II
COMPLEXITY COMPARISON OF THREE DIFFERENT DYNAMIC PRICING ALGORITHMS.

	Computational complexity	Memory complexity
Q-learning with original state	$O(\|\mathcal{A}\|)$	$O(\prod_{i \in \mathcal{I}} \|\mathcal{D}_i\| \|\mathcal{H}\| \|\mathcal{C}\| \|\mathcal{A}\|)$
Q-learning with EAS	$O(\|\mathcal{A}\|)$	$O(\|\mathcal{E}\| \|\mathcal{H}\| \|\mathcal{C}\| \|\mathcal{A}\|^2)$
Q-learning with EAS and virtual experience	$O(\|\theta(\hat{\sigma})\| \|\mathcal{A}\|)$	$O(\|\mathcal{E}\| \|\mathcal{H}\| \|\mathcal{C}\| \|\mathcal{A}\|^2)$

IV. Reinforcement Learning at Customers: Energy Consumption Scheduling Algorithm

In the previous sections, we assumed that each customer's energy consumption at each time-slot is determined in a myopic way by which the customer minimizes its cost at the current time-slot without considering its influence on its expected cost. However, with an appropriate learning algorithm in each household, each customer also can foresee

[4] Note that this is a reasonable assumption because the service provider can gather sufficient data to estimate the transition probability of the cost function since it participates in the market for a long time. We leave the discussion on the errors on estimate of the cost function and its impact on the system performance for future work.

[5] It is worth noting that the number of virtual experience tuples strongly depends on the model of electricity cost, i.e., the number of cost functions and its state transition probability $p_c(c^{t+1}|c^t)$. However, we do not focus on how specific types of cost dynamic model influence the system performance.

the change of the retail price and take this into account in its decision making based on the observation of the service provider's decision on the retail price. This learning capability of the customers will help themselves to further minimize their expected costs as well as the system cost for the service provider.

In this section, we extend our system model to the multi-agent microgrid system where not only the service provider but also each customer can exploit the learning capability in its decision making. Each customer aims at minimizing its expected cost and decides its energy consumption based on the observed retail price. We then propose a reinforcement learning algorithm by which each customer can determine its energy consumption in a distributed manner without a priori information exchange with the service provider or other customers.

A. System Model and Problem Formulation

In the multi-agent microgrid system, similarly to the service provider, each customer i can conduct learning on its energy consumption strategies. As a decision maker, each customer chooses an energy consumption function a_i^t at each time-slot t among the set of energy consumption functions $\mathcal{A}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,A_i}\}$. Then, the actual energy consumption of customer i, e_i^t, is calculated based on the energy consumption function a_i^t and the current accumulated load demand d_i^t, i.e,

$$e_i^t = a_i^t(d_i^t). \tag{16}$$

Each customer i decides its energy consumption function a_i^t based on the observation of its state s_i^t which is defined as a combination of its current accumulated load demand d_i^t, the current period h^t, and the current retail price a^t, i.e.,

$$s_i^t = (d_i^t, h^t, a^t) \in \mathcal{S}_i, \tag{17}$$

where $\mathcal{S}_i = \mathcal{D}_i \times \mathcal{H} \times \mathcal{A}$ is the set of customer i's states. We denote customer i's stationary policy that maps its states $\mathcal{S}_i$ to the actions $\mathcal{A}_i$ by $\pi_i : \mathcal{S}_i \rightarrow \mathcal{A}_i$, i.e., $a_i^t = \pi_i(s_i^t)$. In the multi-agent microgrid system, we assume that each customer can learn to minimize its expected long-term (discounted) cost by solving the following problem:

$$\mathbf{P}_i : \min_{\pi_i : \mathcal{S}_i \rightarrow \mathcal{A}_i} E\left[\sum_{t=0}^{\infty} (\gamma)^t \phi_i^t(d_i^t, e_i^t)\right], \quad \forall i \in \mathcal{I}. \tag{18}$$

For the service provider, we let it solve the same MDP problem $\mathbf{P}$ by using the reinforcement learning-based dynamic pricing algorithm proposed in Section III, except that the customers' energy consumption $e_i^t, \forall i \in \mathcal{I}$ described in (12) is replaced by the output of the energy consumption scheduling algorithm which will be developed through this section.

B. Post-Decision State Learning

To solve each customers's problem, we can use conventional reinforcement learning algorithms such as Q-learning, however, they do not exploit the known information about the system and this may limit its learning speed. In practice, the customers have partial knowledge about the transition of the system states, but the retail price a^{t+1} at the next time-slot is unknown to the customer in advance. We can categorize the information about the microgrid into a known part and an unknown part according to whether each customer can obtain it or not before its decision is made, as shown in Table III. We assume that customer i knows the distribution of its new load demand at the next time-slot, D_i^{t+1}.

TABLE III
KNOWN AND UNKNOWN INFORMATION IN CUSTOMER'S PROBLEM

Known information
• Period at next time-slot, $h^{t+1} = \mod(t+1, H)$ • Backlog and extended demand, $\lambda_i(d_i^t - e_i^t) + \mu_i e_i^t$ • Cost for current time-slot, $\phi_i^t(d_i^t, e_i^t) = u_i(d_i^t - e_i^t) + a^t(e_i^t)$ • Transition probability of load demand level at next time-slot, $p_{d_i}(d_i^{t+1} \mid d_i^t, a_i^t)$
Unknown information
• Retail price at next time-slot, a^{t+1}

In order to exploit this known information, we apply the concept of the post-decision state (PDS) [29] and develop the corresponding PDS learning algorithm. By exploiting the known information about the system, the PDS learning algorithm can remove the need for action exploration and improve the learning speed.

We first define customer i's PDS as the state where the known information is reflected based on customer i's decision on a_i^t, but the unknown information is not reflected. Accordingly, we denote customer i's PDS at time-slot t by $\bar{s}_i^t = (\bar{d}_i^t, \bar{h}^t, a^t) \in \mathcal{S}_i$ [6], where $\bar{d}_i^t = d_i^{t+1}$ and $\bar{h}^t = h^{t+1}$. Based on the PDS, the state transition from state s_i^t to s_i^{t+1} can be represented as:

- State at time-slot t: $s_i^t = (d_i^t, h^t, a^t)$
- PDS at time-slot t: $\bar{s}_i^t = (d_i^{t+1}, h^{t+1}, a^t)$
- State at time-slot $t+1$: $s_i^{t+1} = (d_i^{t+1}, h^{t+1}, a^{t+1})$

Then, we can represent the state transition probability from s_i^t to s_i^{t+1} as

$$p_{s_i}(s_i^{t+1} \mid s_i^t, a_i^t) = \sum_{\bar{s}_i \in \mathcal{S}_i} p_k(\bar{s}_i \mid s_i^t, a_i^t) p_u(s_i^{t+1} \mid \bar{s}_i), \tag{19}$$

where $p_u(s_i^{t+1} \mid \bar{s}_i^t)$ and $p_k(\bar{s}_i^t \mid s_i^t, a_i^t)$ denote the unknown and known probabilities, respectively. Since the period update process is deterministic, the known probability can be represented as $p_k(\bar{s}_i^t \mid s_i^t, a_i^t) = p_d(d_i^{t+1} \mid d_i^t, a_i^t)$. Since each customer can calculate its cost before it decides the energy consumption function, customer i's cost consists of only the known part, i.e.,

$$\begin{aligned} \phi(s_i^t \mid a_i^t) &= \phi_i^t(d_i^t, e_i^t) \\ &= u_i(d_i^t - e_i^t) + a^t(e_i^t). \end{aligned} \tag{20}$$

To develop the PDS learning algorithm, we define the state-

[6] The set of PDSs is the same as that of the customer i's states.

Algorithm 2 PDS Learning Algorithm

1: Initialize $\bar{V}$ arbitrarily, $t = 0$
2: **for each time-slot** t
3: Choose a^t according to policy $\bar{\pi}(s_i^t)$
4: Take action a^t, observe cost $\phi_i^t(d_i^t, e_i^t)$, PDS $\bar{s}_i^t$, and next state s_i^{t+1}
5: $V(s_i^{t+1}) = \min_{a_i \in \mathcal{A}_i} \left[\phi(s_i^{t+1}, a_i) + \sum_{\bar{s}_i \in \mathcal{S}_i} p_k(\bar{s}_i | s_i^{t+1}, a_i) \bar{V}(\bar{s}_i)\right]$
6: $\bar{V}(\bar{s}_i^t) \leftarrow (1-\epsilon)\bar{V}(\bar{s}_i^t) + \epsilon\gamma V(s_i^{t+1})$
7: **end**

value functions of customer i's state and PDS as

$$\bar{V}^*(\bar{s}_i) = \gamma \sum_{s_i' \in \mathcal{S}_i} p_u(s_i' | \bar{s}_i, a_i) V^*(s_i') \tag{21}$$

$$V^*(s_i) = \min_{a_i \in \mathcal{A}_i} \left[\phi(s_i, a_i) + \sum_{\bar{s}_i \in \mathcal{S}_i} p_k(\bar{s}_i | s_i, a_i) \bar{V}^*(\bar{s}_i)\right], \tag{22}$$

respectively. Given the optimal PDS value function, the optimal stationary policy can be computed as

$$\bar{\pi}_i^*(s_i) = \min_{a_i \in \mathcal{A}_i} \left[\phi(s_i, a_i) + \sum_{\bar{s}_i \in \mathcal{S}_i} p_k(\bar{s}_i | s_i, a_i) \bar{V}^*(\bar{s}_i)\right]. \tag{23}$$

We outline the PDS learning algorithm in Algorithm 2. The PDS learning algorithm can improve the learning speed compared to the conventional Q-learning algorithm by taking advantages of the following characteristics:

- The update process of value functions $V(s_i^{t+1})$ and $\bar{V}(s_i^t)$ in lines 5-6 of Algorithm 2 provides information about the state-value function of many states. This is possible thanks to the exploitation of the known parts of probability $p_k(\bar{s}_i^t | s_i^t, a_i^t)$ and cost $\phi(s_i, a_i)$.
- The PDS learning algorithm can choose its action in a greedy manner at each time-slot as in line 4 of Algorithm 2 without requiring the randomized exploration in each state.

The computational complexity and the memory complexity of the proposed energy consumption scheduling algorithm are summarized in Table IV. The computational complexity of the PDS learning algorithm at each time-slot is larger than the conventional Q-learning algorithm since the learning update of PDS learning algorithm spans all $|S_i|$ states for each action a_i. However, PDS learning algorithm leads to less memory complexity than the conventional Q-learning algorithm by storing only the state-value functions for the original states $V(s_i)$ and those of PDSs $\bar{V}(\bar{s}_i)$ without requiring to store the action-value functions $Q(s_i, a_i)$.

V. Numerical Results

In this section, we provide numerical results to evaluate the performance of our dynamic pricing algorithm. One day consists of 24 time-slots each of which lasts for one hour.

We consider a microgrid with 20 customers. We assume that the newly generated load demand of customer i, D_i^t, follows a Poisson distribution [7] with expected value ω_{i,h^t}, which is proportional to the hourly average load shapes of residential electricity services in California [32] as shown in Fig. 3. We assume that all customers have the same backlog rate, i.e., $\lambda_i = \lambda, \forall i \in \mathcal{I}$, but the demand extension rate μ_i is uniform randomly determined at every time-slot among values in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

TABLE IV
Complexity comparison of two different energy consumption scheduling algorithms.

	Computational complexity	Memory complexity
Q-learning	$O(\lvert\mathcal{A}\rvert)$	$O(\lvert\mathcal{D}_i\rvert\lvert\mathcal{H}\rvert\lvert\mathcal{A}\rvert\lvert\mathcal{A}_i\rvert)$
PDS learning	$O(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)$	$O(2\lvert\mathcal{D}_i\rvert\lvert\mathcal{H}\rvert\lvert\mathcal{A}\rvert)$

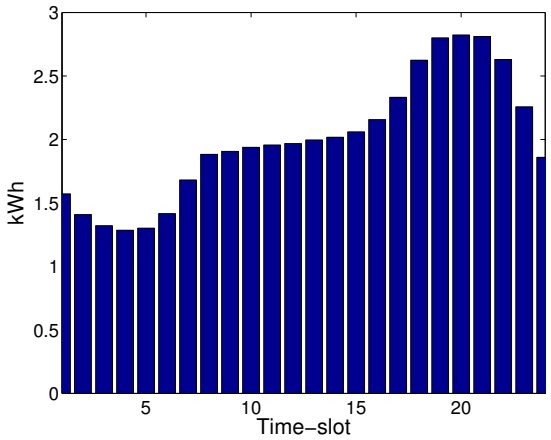

Fig. 3. Load demand profile.

Each customer i's disutility function $u_i(d_i^t - e_i^t)$ is given as

$$u_i(d_i^t - e_i^t) = \kappa_i \times (d_i^t - e_i^t)^2, \tag{24}$$

where κ_i is a constant that represents customer i's disutility sensitivity to its remaining demand. Here, we let $\kappa_i = \kappa = 0.1, \forall i \in \mathcal{I}$. If not specified, we only consider the customers without learning capabilities, i.e., each customer decides its energy consumption in a myopic manner to minimize its current cost. We consider the impact of customers' learning capability in Subsection V-D.

We model the cost function c^t as a quadratic function [8] of the total energy consumption $\sum_{i \in \mathcal{I}} e_i^t$ as in [6][15][16][19][33][34][35][36] :

$$c^t\left(\sum_{i \in \mathcal{I}} e_i^t\right) = \alpha^t \times \sum_{i \in \mathcal{I}} e_i^t + \beta_{h^t}^t \times \left(\sum_{i \in \mathcal{I}} e_i^t\right)^2. \tag{25}$$

We set $\alpha^t = 0.02, \forall t$ and $\beta_{h^t}^t$ to be a random variable whose expected value, v_{h^t}, changes according to the corresponding period h^t based on the hourly average load shape in Fig.

[7] The Poisson distribution of load demand has been adopted in many existing works such as [23],[30],[31]. Note also that our reinforcement learning algorithms are able to operate efficiently regardless of the types of the demand arrival model.

[8] Note that the choice of the cost function in this section is only an illustrative example and our reinforcement learning algorithms are able to operate efficiently regardless of the cost function.

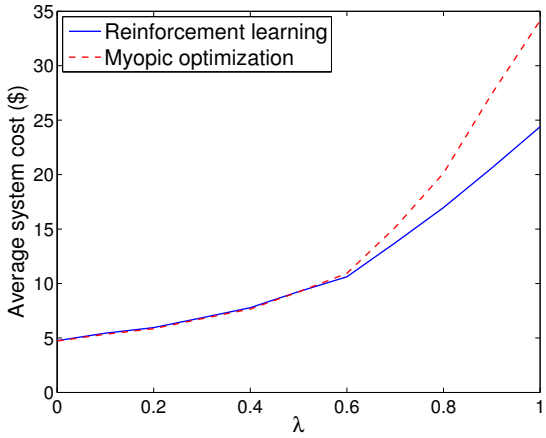

Fig. 4. Performance comparison of our reinforcement learning algorithm and the myopic optimization algorithm varying λ.

3. With a given period h^t, $\beta^t_{h^t}$ is uniform randomly chosen among values in $\{0.25v_{h^t}, 0.5v_{h^t}, \cdots, 1.75v_{h^t}\}$. The discount factor γ is fixed to 0.95 in problems $\mathbf{P}$ and $\mathbf{P}_i, \forall i$. The retail pricing function a^t is a linear function of the energy consumption e^t_i, i.e.,

$$a^t(e^t_i) = \chi^t e^t_i, \tag{26}$$

where the coefficient χ^t can be chosen among set $\{0.2, 0.4, \cdots, 1.0\}$ each element of which is directly mapped to one retail pricing function in $\mathcal{A}$.

A. Performance Comparison with Myopic Optimization

We first evaluate the performance of our pricing algorithm by comparing it with that of *myopic optimization algorithm*. In the myopic optimization algorithm, the service provider chooses an action with the lowest expected instantaneous system cost, which can be updated similarly to the Q-learning update by letting the discount factor $\gamma = 0$. This implies that the myopic optimization algorithm focuses only on the immediate system cost without considering the impact of the current action on the future system cost. In Fig. 4, we show the average system costs of those two pricing algorithms by changing the backlog rate, λ, from 0 to 1. We set $\rho = 0.5$. We can observe that the average system costs increase as λ increases in both pricing algorithms because with a higher backlog rate, the accumulated load demand causes a higher disutility. We also observe that the performance gap between two algorithms increases as λ increases. For example, with a zero backlog rate, $\lambda = 0$, the solution of our reinforcement learning algorithm is the same as that of the myopic optimization algorithm. On the other hand, in the case with a higher backlog rate, the remaining backlog is carried forward to the next time-slot. Hence, the service provider's pricing decision at a time-slot influences the accumulated load demand in the future, and thus its future system cost. Due to this difference, in dynamic pricing, the ability to forecast the future system cost is more important especially when λ is large.

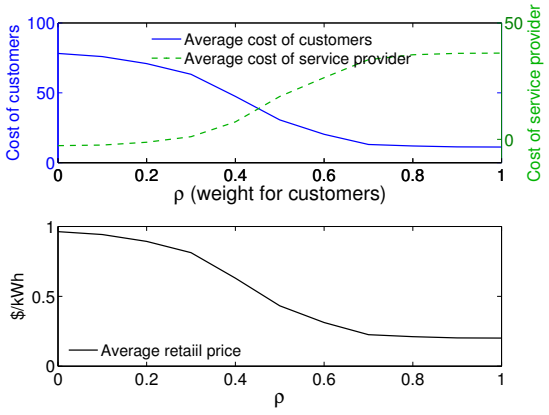

Fig. 5. Impact of the weighting factor ρ on the performances of customers and service provider.

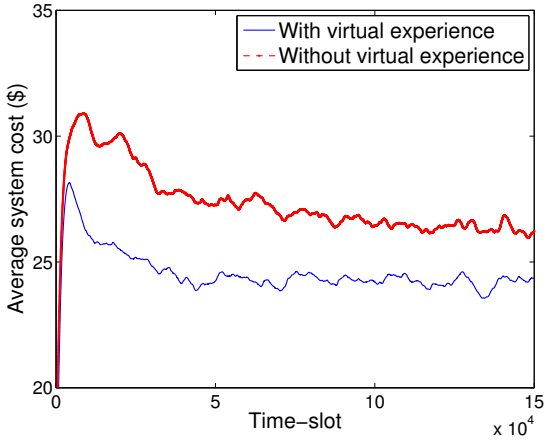

Fig. 6. Impact of virtual experience on the learning speed of the Q-learning algorithm.

B. Impact of Weighting Factor ρ

To study the impact of the weighting factor ρ, in Fig. 5, we show the cost of customers, that of the service provider, and the average retail price with varying ρ from 0 to 1. We set $\lambda = 1$. We can observe that as ρ increases, the service provider reduces the average retail price, the cost of customers decreases, and the cost of the service provider increases. For example, in the case with $\rho = 0$, the service provider aims at minimizing its own cost. Hence, the service provider does not consider the customers' disutility and chooses relatively high prices to reduce the its own cost. On the other hand, in the case with $\rho = 1$, the service provider aims at minimizing the customers' cost. Hence, the service provider chooses relatively low prices to provide electric energy to the customers at a low retail price as possible.

C. Virtual Experience Update

In Fig. 6, we compare the learning speed of our virtual experience-based reinforcement learning algorithm with that of the conventional Q-learning algorithm without virtual experience. We set $\lambda = 1$ and $\rho = 0.5$. We can observe that our algorithm with virtual experience provides a significantly improved learning speed compared to that of the conventional

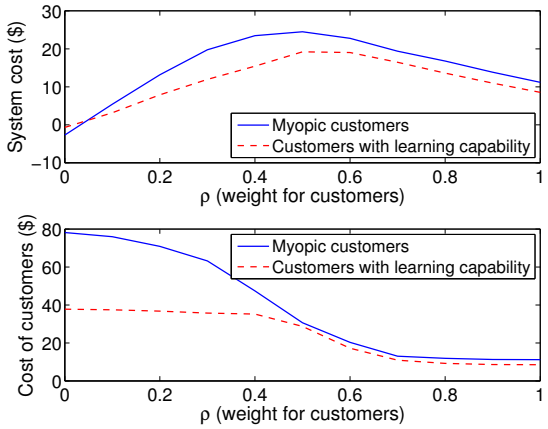

Fig. 7. Impact of the customers' learning capability.

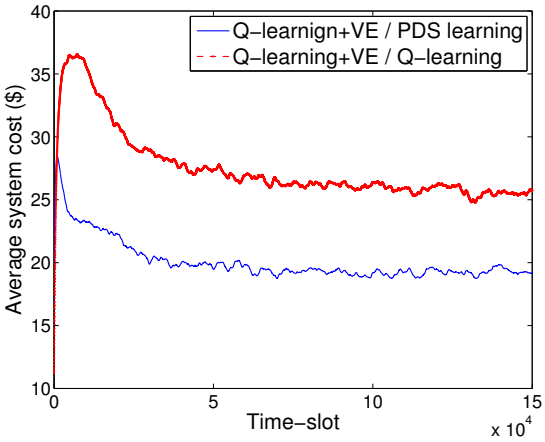

Fig. 8. Comparison of learning speed of the PDS learning algorithm and the conventional Q-learning algorithm.

Q-learning algorithm. This means that even if the stochastic characteristic of the system dynamics vary in time, the proposed reinforcement learning algorithm can quickly adapt to the time-varying environment by exploiting virtual experience update.

D. Customers with Learning Capability

We now study the impact of customers' learning capability on the performance of the microgrid. For each customer i's problem $\mathbf{P}_i$, we use the PDS learning algorithm in Algorithm 2. The energy consumption function a_i^t is a linear function of customer i's accumulated load demand d_i^t, i.e.,

$$a_i^t(d_i^t) = \chi_i^t d_i^t, \tag{27}$$

where the coefficient χ_i^t can be chosen from set $\{0, 0.2, \cdots, 1.0\}$ each element of which is directly mapped to one energy consumption function in $\mathcal{A}_i$. We use the same simulation environment as in the previous subsections.

In Fig. 7, we compare the performances of two different scenarios with varying ρ from 0 to 1: one has the customers with the learning capability and the other has the myopic customers. For both scenarios, we set $\lambda = 1$. Fig. 7 shows that the customers with learning capability results in the lower average system cost as well as the lower customers' average cost than the myopic customers in most environments. It is worth noting that with a small ρ, the customers with the learning capability achieves a significantly reduced cost than the myopic customers. With a small ρ, the service provider places more importance on its own cost rather than that of the customers, and thus sets the retail price as high as possible to maximize its income from the electricity bills. However, the result on the cost of customers show that the customers with the learning capability can cope with the service provider's unilateral decision on the retail price by efficiently scheduling its energy consumption.

In Fig. 8, we evaluate the learning speed of the proposed PDS learning algorithm. For the service provider's dynamic pricing, we use the proposed Q-learning algorithm with virtual experience (Q-Learning+VE). For each customer's energy consumption scheduling, we compare the proposed PDS learning algorithm with the conventional Q-learning algorithm. We set $\lambda = 1$ and $\rho = 0.5$. Fig. 8 shows that thanks to the learning capability of customers, the lower average system cost is achieved compared to the result shown in Fig. 6. As multiple agents including the customers and the service provider individually perform reinforcement learning algorithms at the same time, the learning speed is extremely slow when conventional Q-learning algorithm is used. However, the proposed PDS learning algorithm provides a significantly improved learning speed by exploiting the known information at each customer.

VI. Conclusion

In this paper, we studied a dynamic pricing problem for the microgrid system where the service provider can adaptively decide the electricity price according to the customers' load demand levels and the cost. In developing the reinforcement learning-based dynamic pricing algorithm, to resolve the existing drawbacks of the conventional reinforcement learning algorithm, we proposed two improvements: energy consumption-based approximate state (EAS) definition and the adoption of virtual experience update in the conventional Q-learning algorithm. We then study a more intelligent microgrid system where each customer to learn and foresee the change of retail price. By adopting the multi-agent learning structure with the PDS learning algorithm, we develop the distributed algorithm which can operate at each of the service provider and the customers without a priori information exchange. Numerical results show that the reinforcement learning-based dynamic pricing achieves a higher long-term performance compared to the myopic optimization approach especially in the system where the customers have a high demand backlog rate. We also showed that by utilizing the customers' learning capability, we can significantly reduce the customers' cost as well as the system cost. Moreover, the improved learning speed of our algorithms with the alternative state definition, virtual experience, and PDS learning enables the proposed reinforcement learning algorithms to be conducted online in a fully distributed manner.

The results in this paper can be extended in several directions. First, in Section IV, we used a straightforward approach by which each customer or the service provider regards the other agent's decision as a part of the environment and learn their behaviors. This approach is reasonable because each of many customers in the microgrid is not necessarily strategic; instead, it is more important to learn the dynamics of the entire system and find its optimal energy consumption scheduling based on the observations. Even though we showed that our approach achieves good performances in terms of both convergence and system cost, there exist a potential for further improvements by studying the strategic behaviors of the rational agents and its impact on the system performance, which is an interesting future direction of this paper. Second, in this paper, we assumed that the service provider knows the transition probability of cost functions. However, in practice, there might exist some errors on the estimates of the electricity cost and this would affect the performance of the proposed reinforcement learning algorithm. We are planning to discuss and show this impact in future work. Third, in this paper, we did not explicitly model the various types of energy sources, e.g., solar power, and energy consumers, e.g., electric vehicles. It will be interesting to take into account the impact of the various types of energy sources/consumers on the dynamic pricing policies and energy consumption scheduling as well as the impact of the bidirectional energy delivery and different types of pricing structures between the service provider and the customers as introduced in [37].

REFERENCES

[1] B.-G. Kim, Y. Zhang, M. van der Schaar, and J.-W. Lee, "Dynamic pricing for smart grid with reinforcement learning," in *IEEE CCSES (IEEE INFOCOM Workshop)*, 2014.

[2] M. H. Albadi and E. El-Saadany, "Demand response in electricity markets: An overview," in *IEEE Power Engineering Society General Meeting*, 2007.

[3] J.-W. Lee and D.-H. Lee, "Residential electricity load scheduling for multi-class appliances with time-of-use pricing," in *IEEE GLOBECOM Workshops*, 2011.

[4] H.-T. Roh and J.-W. Lee, "Residential demand response scheduling for multi-class appliances in the smart grid," *IEEE Transactions on Smart Grid*, to appear. [Online]. Available: http://dx.doi.org/10.1109/TSG.2015.2445491

[5] F. Ruelens, B. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets, and R. Belmans, "Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning," in *Power Systems Computation Conference (PSCC)*, 2014.

[6] P. Samadi, R. Schober, and V. Wong, "Optimal energy consumption scheduling using mechanism design for the future smart grid," in *IEEE SmartGridComm*, 2011.

[7] B.-G. Kim, S. Ren, M. van der Schaar, and J.-W. Lee, "Bidirectional energy trading and residential load scheduling with electric vehicles in the smart grid," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1219–1234, July 2013.

[8] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 120–133, Sept. 2010.

[9] P. Yi, X. Dong, A. Iwayemi, C. Zhou, and S. Li, "Real-time opportunistic scheduling for residential demand response," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 227–234, March 2013.

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

[11] S.-J. Kim and G. Giannakis, "Scalable and robust demand response with mixed-integer constraints," *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2089–2099, Dec. 13.

[12] Y. Zhang and M. van der Schaar, "Structure-aware stochastic storage management in smart grids," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1098–1110, Dec. 2014.

[13] Z. Wen, D. O'Neill, and H. R. Maei, "Optimal demand response using device based reinforcement learning," *abs/1401.1549*, 2014. [Online]. Available: I://arxiv.org/abs/1401.1549

[14] E. Kara, M. Berges, B. Krogh, and S. Kar, "Using smart devices for system-level management and control in the smart grid: A reinforcement learning framework," in *IEEE SmartGridComm*, 2012.

[15] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *IEEE SmartGridComm*, 2010.

[16] P. Tarasak, "Optimal real-time pricing under load uncertainty based on utility maximization for smart grid," in *IEEE SmartGridComm*, 2011.

[17] C. Joe-Wong, S. Sen, S. Ha, and M. Chiang, "Optimized day-ahead pricing for smart grids with device-specific scheduling flexibility," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1075–1085, July 2012.

[18] L. P. Qian, Y. Zhang, J. Huang, and Y. Wu, "Demand response management via real-time electricity price control in smart grids," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1268–1280, July 2013.

[19] C. Chen, S. Kishore, and L. Snyder, "An innovative RTP-based residential power scheduling scheme for smart grids," in *IEEE ICASSP*, 2011.

[20] P. Yang, G. Tang, and A. Nehorai, "A game-theoretic approach for optimal time-of-use electricity pricing," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 884–892, May 2013.

[21] J. Ma, J. Deng, L. Song, and Z. Han, "Incentive mechanism for demand side management in smart grid using auction," *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1379–1388, May 2014.

[22] B. Chai, J. Chen, Z. Yang, and Y. Zhang, "Demand response management with multiple utility companies: A two-level game approach," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 722–731, March 2014.

[23] M. He, S. Murugesan, and J. Zhang, "Multiple timescale dispatch and scheduling for stochastic reliability in smart grids with wind generation integration," in *IEEE INFOCOM*, 2011, pp. 461–465.

[24] Q. Huang, M. Roozbehani, and M. A. Dahleh, "Efficiency-risk tradeoffs in dynamic oligopoly markets – with application to electricity markets," in *IEEE CDC*, 2012.

[25] M. Roozbehani, A. Faghih, M. I. Ohannessian, and M. A. Dahleh, "The intertemporal utility of demand and price elasticity of consumption in power grids with shiftable loads," in *IEEE CDC-ECC*, 2011.

[26] B. Asare-Bediako, W. Kling, and P. Ribeiro, "Integrated agent-based home energy management system for smart grids applications," in *2013 4th IEEE/PES ISGT EUROPE*, 2013.

[27] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[28] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, 1989.

[29] N. Mastronarde and M. van der Schaar, "Joint physical-layer and system-level power management for delay-sensitive wireless communications," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 694–709, April 2013.

[30] G. Koutitas, "Control of flexible smart devices in the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1333–1343, Sept. 2012.

[31] S. Yue, J. Chen, Y. Gu, C. Wu, and Y. Shi, "Dual-pricing policy for controller-side strategies in demand side management," in *IEEE SmartGridComm*, 2011.

[32] *Dynamic Load Profiles in California*. Pacific Gas & Electric. [Online]. Available: http://www.pge.com/tariffs/energy_use_prices.shtml

[33] A. Mohsenian-Rad, V. 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 Transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[34] Y. Huang, S. Mao, and R. M. Nelms, "Adaptive electricity scheduling in microgrids," in *IEEE INFOCOM*, 2013.

[35] C. Lin and G. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 6, pp. 1170–1175, June 1984.

[36] M. Tushar, C. Assi, M. Maier, and M. Uddin, "Smart microgrids: Optimal joint scheduling for electric vehicles and home appliances," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 239–250, Jan 2014.

[37] R. Verzijlbergh and Z. Lukszo, "Conceptual model of a cold storage warehouse with pv generation in a smart grid setting," in *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2013.