Adaptive Contextual Learning for Unit Commitment in Microgrids With Renewable Energy Sources

Hyun-Suk Lee, Cem Tekin, *Member, IEEE*, Mihaela van der Schaar, *Fellow, IEEE*, and Jang-Won Lee, *Senior Member, IEEE*

***Abstract*—In this paper, we study a unit commitment (UC) problem where the goal is to minimize the operating costs of a microgrid that involves renewable energy sources. Since traditional UC algorithms use *a priori* information about uncertainties such as the load demand and the renewable power outputs, their performances highly depend on the accuracy of the *a priori* information, especially in microgrids due to their limited scale and size. This makes the algorithms impractical in settings where the past data are not sufficient to construct an accurate prior of the uncertainties. To resolve this issue, we develop an adaptively partitioned contextual learning algorithm for UC (AP-CLUC) that learns the best UC schedule and minimizes the total cost over time in an online manner without requiring any *a priori* information. AP-CLUC effectively learns the effects of the uncertainties on the cost by adaptively considering context information strongly correlated with the uncertainties, such as the past load demand and weather conditions. For AP-CLUC, we first prove an analytical bound on the performance, which shows that its average total cost converges to that of the optimal policy with perfect *a priori* information. Then, we show via simulations that AP-CLUC achieves competitive performance with respect to the traditional UC algorithms with perfect *a priori* information, and it achieves better performance than them even with small errors on the information. These results demonstrate the effectiveness of utilizing the context information and the adaptive management of the past data for the UC problem.**

***Index Terms*—Contextual learning, unit commitment, microgrids, renewable energy, system uncertainty.**

I. Introduction

USING renewable energy sources such as wind and solar has many advantages, e.g., low economic costs and carbon footprint reduction from fossil fuels. In general, to efficiently use renewable energy sources in power systems, uncertainties in power systems from load demands and renewable power outputs should be effectively addressed. Especially, in microgrids, properly addressing the uncertainties becomes more important due to their limited scale and size.

Recently, such uncertainties are considered in the unit commitment (UC) problems to determine the on/off states of the thermal generation units and their power outputs, i.e., the UC schedule, to minimize operating costs. In many existing works [2]–[10], UC schedules that take into account the system uncertainties are determined by using stochastic optimization for UC (SOUC). In SOUC, the UC schedule is determined to minimize the expected operating cost over possible scenarios of the uncertainties. However, in practice, the number of scenarios considered in SOUC should be reduced due to its high computational complexity [11], which causes reliability issues since the reduced scenarios do not capture all possibilities.

To resolve the reliability issues, two different approaches named robust optimization for UC (ROUC) and interval optimization for UC (IOUC) are proposed. In ROUC [12]–[17], the UC schedule is determined to minimize the worst-case cost using a deterministic uncertainty set defined by the worst-case realization. In IOUC [18]–[20], the UC schedule is determined considering reliability constraints defined by using intervals representing the probable realizations of the uncertainties. However, the above approaches have a difficulty in appropriately choosing the scenarios in SOUC, the uncertainty set in ROUC, and the intervals in IOUC to tradeoff the reliability and the costs. To overcome this difficulty, in [21], a Markovian approach for UC is proposed where the UC schedule is determined without scenario analysis by representing the uncertainties using a discrete Markov process. Moreover, numerous works have considered hybrid approaches that combine the base approaches discussed above [22]–[25].

Although many prior works determine UC schedules by considering uncertainties in different ways, they all require statistical information about the realization of the uncertainties as follows: a probability distribution over the scenarios in SOUC, the forecasted worst-case realizations of uncertainties in ROUC, the uncertainty intervals in IOUC, and the stochastic model of the uncertainties in Markovian UC. We call such statistical information about the uncertainties *a priori information*. Due to this dependency, the prior works can be used only if a priori information is available. One way to tackle this issue is to

Manuscript received September 29, 2017; revised April 9, 2018 and June 14, 2018; accepted June 15, 2018. Date of publication June 22, 2018; date of current version July 27, 2018. The work of H.-S. Lee and J.-W. Lee was supported by Midcareer Researcher Program through NRF grant funded by the MSIT, Korea (No. NRF-2017R1A2B4006908). The work of M. van der Schaar was supported in part by an ONR grant and in part by the NSF under Grants 1407712, 1524417, and 1533983. This paper was presented in part at the 5th IEEE Global Conference on Signal and Information Processing, Greater Washington, D.C., Dec. 2016. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Dipti Srinivasan. *(Corresponding author: Jang-Won Lee.)*

H.-S. Lee and J.-W. Lee are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea (e-mail: hs.lee@yonsei.ac.kr; jangwon@yonsei.ac.kr).

C. Tekin is with the Electrical and Electronics Engineering Department, Bilkent University, Ankara 06800, Turkey (e-mail: cemtekin@ee.bilkent.edu.tr).

M. van der Schaar is with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2018.2849855

TABLE I
COMPARISONS WITH RELATED WORKS

	Methodology	Load demand uncertainty	Renewable energy uncertainty	A priori information of uncertainties	Theoretical performance bounds
[2], [3], [8]	SOUC	Considered	Not Considered	Probabilities on scenarios	Not provided
[7]	SOUC	Not Considered	Considered	Probabilities on scenarios	Not provided
[4]–[6], [9], [10]	SOUC	Considered	Considered	Probabilities on scenarios	Not provided
[12]–[17]	ROUC	Considered	Considered	Forecasted values on uncertainties	Not provided
[18], [19]	IOUC	Considered	Considered	Uncertainty intervals	Not provided
[20]	IOUC	Not Considered	Considered	Uncertainty intervals	Not provided
[21]	Markovian UC	Considered	Considered	Markov process for uncertainties	Not provided
[22]	SOUC+ROUC	Considered	Not Considered	Probabilities on scenarios & forcasted values on uncertainties	Not provided
[23], [24]	SOUC+IOUC	Considered	Considered	Probabilities on scenarios	Not provided
[25]	Markovian UC+IOUC	Considered	Considered	Markov process for uncertainties	Not provided
Our works	Contextual learning	Considered	Considered	Not needed	Provided

form the a priori information by acquiring more data, which costs both money and time. Then, this data can be processed by appropriate methods [26], [27] to form estimates of the uncertainties. In addition to the money and time costs, this method also has the following drawback: the performance of methods that are based on a priori information highly deteriorate when the a priori information is inaccurate.

The above approaches for UC are also widely adopted in microgrids [7]–[10], [16]–[18]. However, when adopting the approaches in microgrids, acquiring such an accurate a priori information may cost too much considering their small-scale power generation. Moreover, due to the limited scale and size of microgrids, the performance deterioration from the inaccurate a priori information may become severe [9], [28]. Thus, to overcome these problems, a UC algorithm which does not need any a priori information of uncertainties is necessary, especially in microgrids. The problems related to the a priori information also arise in smart grids [29]–[32], and are addressed by using learning methods that do not require a priori information [31], [32].

To effectively determine UC schedules even without the a priori information, it is necessary to exploit side information strongly correlated with the uncertainties, such as the past load demands and the weather [33], [34]. In the literature, such side information is also referred to as the context information, and the learning methods that utilize the context information are called contextual learning methods [35]. While contextual learning methods are successfully applied in domains like recommender systems [36] and wireless communications [37], to the best of our knowledge, this paper is the first to attempt to use contextual learning for developing a UC algorithm that does not require any a priori information on the uncertainties.

In our preliminary work [1], we developed a uniformly partitioned contextual learning algorithm for UC (UP-CLUC), where the expected costs of the UC schedules are learned by fusing the past data through *uniform* partitioning of the context space. The partition of the context space of UP-CLUC is optimized under the condition that the contexts are uniformly distributed over the context space. This might pose a significant performance degradation in real-world scenarios where the context arrivals are non-uniform or do not follow any well defined stochastic process. To address this challenge, in this paper we propose a contextual learning algorithm for the UC problem called an adaptive partitioned contextual learning algorithm for UC (AP-CLUC). The algorithm addresses the challenge by learning the uncertainties in a completely adaptive way by forming the context space partition on-the-fly based on the context arrivals observed so far. By this, AP-CLUC optimizes its context space to tradeoff estimation errors and approximation errors that occur during learning. A comparison of our work with the related works is given in Table I.

The contributions of the paper are summarized as follows:

- We propose a new contextual learning UC approach without requiring any a priori information by modeling the UC problem as a sequential decision making problem and developing a contextual learning algorithm for the problem.
- The developed algorithm called AP-CLUC adaptively partitions the context space to effectively learn about the system uncertainties based on past data. Moreover, we propose methods to accelerate the learning speed of AP-CLUC.
- We prove that AP-CLUC achieves regret which is sublinear in time, and hence, is optimal in terms of the long-term average cost.
- We also show that AP-CLUC achieves competitive performance compared to the existing UC algorithms having perfect a priori information, and it achieves better performance than them even when the a priori information has only small errors.

The rest of this paper is organized as follows. Section II provides the system model. In Section III, we formulate a unit commitment problem. In Section IV, we develop an adaptively partitioned contextual learning algorithm for UC, and provide its regret bound. We provide numerical results in Section V. Finally, we conclude in Section VI.

II. SYSTEM MODEL

We consider the UC problem in an isolated microgrid system with J thermal power generation units, where each unit is indexed by $j \in \mathcal{J} = \{1, 2, ..., J\}$.[1] The system schedules the on/off status and power outputs of its thermal power generation units, i.e., a UC schedule, over a discrete time horizon, where each time period has a fixed duration, e.g., an hour. Let t be an index of time periods of the time horizon. The set of time periods

[1] In the following, unit j implies thermal power generation unit j.

is denoted by $\mathcal{T} = \{0, 1, 2, ...\}$. At the beginning of time period t, the system schedules its thermal power generation units for a single time period $t + T_{sc}$, i.e., T_{sc} *time periods-ahead UC scheduling*, where T_{sc} is the number of necessary time periods to prepare the operation of the thermal power generation units according to the UC schedule.

The on/off status of unit j during time period t is denoted by $u_j(t) \in \{0, 1\}$, where 1 represents the on state and 0 represents the off state. The vector of the on/off states of all thermal power generation units during time period t is denoted by $\mathbf{u}(t) = \{u_j(t)\}_{j\in\mathcal{J}}$. The up time of unit j at time period t, which represents the number of consecutive time periods that unit j has been in the on state at the *end* of time period t, is denoted by $T_{j,on}(t)$, and is given by

$$T_{j,on}(t) = \begin{cases} T_{j,on}(t-1) + 1, & \text{if } u_j(t) = 1 \\ 0, & \text{if } u_j(t) = 0 \end{cases}.$$

Similarly, the down time of unit j at time period t, which represents the number of consecutive time periods that unit j has been in the off state at the *end* of time period t, is denoted by $T_{j,off}(t)$, and it is obtained by

$$T_{j,off}(t) = \begin{cases} T_{j,off}(t-1) + 1, & \text{if } u_j(t) = 0 \\ 0, & \text{if } u_j(t) = 1 \end{cases}.$$

We denote the vectors of $T_{j,on}(t)$'s and $T_{j,off}(t)$'s of all thermal power generation units as $\mathbf{T}_{on}(t) = \{T_{j,on}(t)\}_{j\in\mathcal{J}}$ and $\mathbf{T}_{off}(t) = \{T_{j,off}(t)\}_{j\in\mathcal{J}}$, respectively. When a thermal power generation unit is turned on, it cannot be turned off for a specific number of time periods, i.e., for each unit j,

$$1 \le T_{j,on}(t-1) < MUT_j \Rightarrow u_j(t) = 1, \tag{1}$$

where MUT_j is the minimum up time of unit j. Similarly, when it is turned off, it cannot be turned on for the next specific number of time periods, i.e., for each unit j,

$$1 \le T_{j,off}(t-1) < MDT_j \Rightarrow u_j(t) = 0, \tag{2}$$

where MDT_j is the minimum down time of unit j.

The power output of unit j during time period t is denoted by $p_j(t)$, and it is bounded by $p_j(t) \in [p_j^{\min}, p_j^{\max}]$, where $p_j^{\min}$ and $p_j^{\max}$ are the minimum and maximum power outputs of unit j, respectively. The vector of the power outputs of all thermal power generation units during time period t is denoted as $\mathbf{p}_{ther}(t) = \{p_j(t)\}_{j\in\mathcal{J}}$. Due to the ramp rate limit, the power output of unit j at time period t should satisfy the following constraint:

$$p_j(t-1) - RR_j \le p_j(t) \le p_j(t-1) + RR_j, \tag{3}$$

where RR_j is the ramp rate limit of unit j. Moreover, we consider a spinning reserve requirement in the system. We assume that the spinning reserve is not used for the fluctuation of the load demand, but for more critical situation such as the outage of thermal units. Then, the spinning reserve requirement should be guaranteed as

$$\sum_{j\in\mathcal{J}} u_j(t)\left(p_j^{\max} - p_j(t)\right) \ge SR, \tag{4}$$

where SR is the spinning reserve requirement.

In our system model, we use the current time, the past weather condition, and the past load demands as the context information which the system considers. It is worth noting that any other related information can be used as the context information. To model the current time, we introduce a set of time indices for a circular time duration, e.g., a day, a month, and a year, $\mathcal{H} = \{0, 1, ..., H-1\}$, where each index represents an actual time in the time duration. Then, each time period t is mapped to the corresponding current time index $h(t) \in \mathcal{H}$ as $h(t) = \mathrm{mod}(t, H)$. Let $w(t)$ be the weather condition which is observed by the system at the beginning of time period t. The set of weather conditions is denoted by $\mathcal{W}$, which can be defined by using weather information components such as wind speed, wind direction, temperature, sky cover, and precipitation potential [26], [27]. When defining it, it is necessary to consider the location of the system and the types of renewable sources of the power generation units, such as wind and solar. For example, for a wind farm, it can be defined as $\mathcal{W} = \mathcal{W}_{windspd} \times \mathcal{W}_{windir}$, where $\mathcal{W}_{windspd}$ and $\mathcal{W}_{windir}$ are the set of wind speeds and wind directions, respectively. Note that both continuous and discrete sets can be used for the weather conditions. The load demand during time period t is denoted by $M(t)$ and is assumed to lie in the bounded interval $\mathcal{M} = [M_{\min}, M_{\max}]$, where $M_{\min}$ and $M_{\max}$ are the minimum and maximum load demands, respectively. The sum of power outputs of all renewable power generation units during time period t is denoted by $p_{re}(t) \le p_{re}^{\max}$, where $p_{re}^{\max}$ is the maximum renewable power output. At the *end* of each time period, the realizations of the random variables that represent the uncertain quantities, i.e., the load demand and the power outputs of the renewable power generation units, are observed by the system. We assume that the distribution of the uncertain quantities at each time period depends on the context information observed at the beginning of that time period.

Due to the system uncertainties, the load demand could be shed or the generated power could be curtailed in our system model. Thus, to ensure the power balance on the system, we define load shedding and power curtailment variables which are determined according to the UC schedule and the realization of the uncertainties. The amount of load shedding during time period t, $p_{sh}(t)$, is given by

$$p_{sh}(t) = \left[M(t) - \sum_{j\in\mathcal{J}} p_j(t) - p_{re}(t)\right]^+,$$

where $[\cdot]^+ = \max[0, \cdot]$. Similarly, the amount of power curtailment during time period t, $p_{cu}(t)$, is given by

$$p_{cu}(t) = \left[\sum_{j\in\mathcal{J}} p_j(t) + p_{re}(t) - M(t)\right]^+.$$

Then, the power balance equation during time period t is derived by

$$\sum_{j\in\mathcal{J}} p_j(t) + p_{re}(t) - p_{cu}(t) = M(t) - p_{sh}(t).$$

The total operating cost of the system during time period t, $C_{tot}(t)$, is obtained as

$$C_{tot}(t)=\sum_{j\in\mathcal{J}}\left(C_{j,fu}(t)+C_{j,su}(t)\right)+C_{sh}(t)+C_{cu}(t), \quad (5)$$

where $C_{j,fu}(t)$ is the fuel cost of unit j that supplies power $p_j(t)$ during time period t, $C_{j,su}(t)$ is the start-up cost of unit j at time period t, $C_{sh}(t)$ is the load shedding cost during time period t, and $C_{cu}(t)$ is the power curtailment cost during time period t. The fuel cost can be modeled as a non-linear function of the power output [38] as

$$C_{j,fu}(t)=C_{j,fu}^{(0)}\cdot u_j(t)+C_{j,fu}^{(1)}\cdot p_j(t)+C_{j,fu}^{(2)}\cdot p_j(t)^2, \quad (6)$$

where $C_{j,fu}^{(0)}$, $C_{j,fu}^{(1)}$, and $C_{j,fu}^{(2)}$ are the cost coefficients of unit j. The start-up cost can be modeled as follows [38], [39]:

$$C_{j,su}(t)=CM_j+CSC_j\left\{1-e^{\left(-\frac{T_{j,off}(t-1)}{CST_j}\right)}\right\}, \quad (7)$$

where CM_j is the start-up cost and maintenance cost of unit j, CSC_j is the cold start-up cost of unit j, and CST_j is the cold start-up time of unit j. The load shedding cost during time period t, $C_{sh}(t)$, is given by

$$C_{sh}(t)=LSP\cdot p_{sh}(t), \quad (8)$$

where LSP is the load shedding price. The power curtailment cost during time period t, $C_{cu}(t)$, is given by

$$C_{cu}(t)=PCP\cdot p_{cu}(t), \quad (9)$$

where PCP is the power curtailment price.

III. Unit Commitment Problem

The context that is observed at the *beginning* of time period t is defined by $\mathbf{x}(t):=\{h(t),\mathbf{M}(t,T_M),\mathbf{w}(t,T_W)\}$, where $\mathbf{M}(t,T_M)=\{M(t-1),...,M(t-T_M)\}$ is the vector of load demands of the past T_M time periods and $\mathbf{w}(t,T_W)=\{w(t-1),...,w(t-T_W)\}$ is the vector of weather conditions of the past T_W time periods. The context space is defined by $\mathcal{X}=\mathcal{H}\times\mathcal{M}^{T_M}\times\mathcal{W}^{T_W}$. We denote the dimension of the context space as $D_{\mathcal{X}}$.

Remark 1: We introduce a projection function ϕ which projects the context $\mathbf{x}$ into a low dimensional space. For example, weighted averaging, principal component analysis (PCA), or mutual information-based dimensionality reduction [40] can be used. Note that the projection function helps our algorithm learn faster if necessary.

In addition to the context $\mathbf{x}$, the down time of units, $\mathbf{T}_{off}$, should be considered when choosing the action since the start-up cost in (7) depends on it. For the sake of analysis, we define the bounded down time of unit j at time period t, $\tilde{T}_{j,off}(t)$, bounded by PDT_j, i.e., $\tilde{T}_{j,off}\in\tilde{\mathcal{T}}_{j,off}=\{0,1,...,PDT_j\}$, where PDT_j is the maximum bounded down time of unit j. Note that since the start-up cost becomes almost a constant for large down times, it is enough to consider down times in a bounded region. Then, the bounded down time space is defined by $\tilde{\mathcal{T}}_{off}=\prod_{j\in\mathcal{J}}\tilde{\mathcal{T}}_{j,off}$. We denote the vector of $\tilde{T}_{j,off}(t)$'s of all units as $\tilde{\mathbf{T}}_{off}(t)=\{\tilde{T}_{j,off}(t)\}_{j\in\mathcal{J}}$. Then, we define an extended context at time period t by $\mathbf{z}(t):=\{\mathbf{x}(t),\tilde{\mathbf{T}}_{off}(t+T_{sc}-1)\}$, and define the extended context space by $\mathcal{Z}=\mathcal{X}\times\tilde{\mathcal{T}}_{off}$.

We now define the state for units at the beginning of time period t as $\mathbf{s}(t):=\{\mathbf{u}(t+T_{sc}-1),\mathbf{p}_{ther}(t+T_{sc}-1),\mathbf{T}_{on}(t+T_{sc}-1),\mathbf{T}_{off}(t+T_{sc}-1)\}$ and let $\mathcal{S}$ denote the state space.

At the beginning of each time period t, an action which is denoted by $a(t)=\{\mathbf{u}(t+T_{sc}),\mathbf{p}_{ther}(t+T_{sc})\}$, is chosen from a subset of the action space, which is defined as $\mathcal{A}=\{0,1\}^J\times\prod_{j\in\mathcal{J}}[p_j^{\min},p_j^{\max}]$. The set of available actions at time period t is constrained by the state (unit status) $\mathbf{s}(t)$ at time period t. Thus, the set of feasible actions at time period t with the unit status $\mathbf{s}(t)$, $\mathcal{A}(\mathbf{s}(t))$, is given as

$$\mathcal{A}(\mathbf{s}(t))=\{\{\mathbf{u}(t+T_{sc}),\mathbf{p}_{ther}(t+T_{sc})\}\in\mathcal{A}| \text{ (1), (2), (3) and (4) holds}\}.$$

We denote a UC policy which depends on the extended context $\mathbf{z}(t)$ and the state $\mathbf{s}(t)$ as $\pi:\mathcal{Z}\times\mathcal{S}\rightarrow\mathcal{A}$. For given extended context $\mathbf{z}(t)$ and unit status $\mathbf{s}(t)$, the UC policy π chooses the action denoted by $\pi_{\mathbf{s}(t)}(t,\mathbf{z}(t))$ from the set of feasible actions, $\mathcal{A}(\mathbf{s}(t))$. For convenience, we denote the action for time period t, $\pi_{\mathbf{s}(t)}(t,\mathbf{z}(t))$, as $\pi(t)$. Then, the UC problem is formally defined by the following equation

$$\underset{\pi:\mathcal{Z}\times\mathcal{S}\rightarrow\mathcal{A}}{\operatorname{argmin}}\ \mathbb{E}\left[\lim_{T\rightarrow\infty}\frac{1}{T}\sum_{t=0}^{T}C_{tot}(\pi(t),t)\right], \quad (10)$$

where $C_{tot}(\pi(t),t)$ is the total operating cost during time period t given action $\pi(t)$. Note that unlike the existing UC models, the UC problem in (10) optimizes the UC over the infinite horizon.

IV. Adaptively Partitioned Contextual Learning

In this section, we introduce an online learning algorithm called adaptively partitioned contextual learning algorithm for UC (AP-CLUC), which solves the UC problem in (10) without requiring any a priori information. We describe AP-CLUC and provide a performance bound for it.

A. How AP-CLUC Learns Effectively?

Basically, contextual learning algorithms learn the effects of the system uncertainties on the costs related to the context and the actions, which are much easier to learn than the entire probability distribution of the system uncertainties. Specifically, to learn the effects, the algorithms estimate the cost of each action with a given context arrival, i.e., an observed context, using the past observed costs of the chosen action with the given context. Then, they learn the best action with the given context arrival by using the estimates of the costs, instead of using a priori information. Hence, they do not require any a priori information. However, when the context space is an uncountable set, the algorithms cannot learn the best action for all possible context arrivals and should approximate context arrivals by merging context arrivals. Thus, in the algorithms, an approximation error from merging context arrivals occurs as well as an estimation error on a cost from limited observations [41].

UP-CLUC in our preliminary work [1] and AP-CLUC partition the context space into multiple sets. Then, they approximate

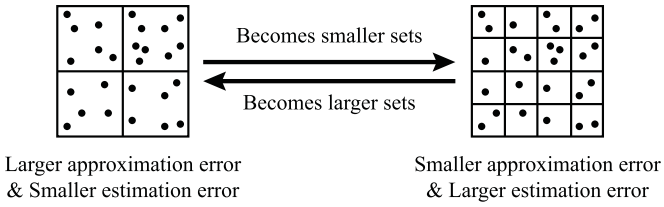

Fig. 1. Illustration of a tradeoff between the estimation error and the approximation error according to a size of sets in the partition of the context space. In the partition, each dot represents each context arrival and each square represents each set.

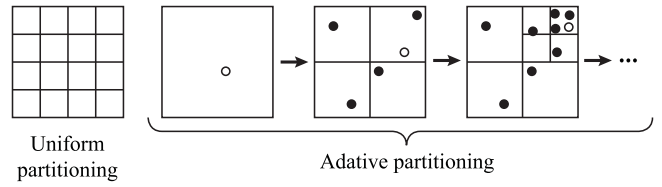

Fig. 2. Illustration of uniform partitioning and adaptive partitioning. In the adaptive partitioning, the filled dots denote the past arrived contexts, and the unfilled dots denote new context arrivals. As the contexts arrive, the context space is adaptively partitioned more precisely.

context arrivals by merging the context arrivals in each set. For such algorithms with a partition-based approximation, the approximation error of each set mainly depends on the size of the set, since the larger size of the set implies that the context arrivals in the larger region are merged. On the other hand, the estimation error of each set depends on the number of context arrivals in the set, since the larger number of context arrivals implies the more accurate estimation. Thus, as shown in Fig. 1, as the size of each set becomes small, the approximation error of each set decreases, but at the same time, the estimation error of each set increases since the number of context arrivals in the set decreases in general. On the other hand, as the size of each set becomes large, the approximation error increases while the estimation error decreases. This results in a tradeoff between the estimation error and the approximation error. Thus, for effective learning, it is important to address the tradeoff considering the context arrivals.

In UP-CLUC, the context space is uniformly partitioned before running, and the size of sets in the context space is determined by a system parameter. This results in the approximation error of the algorithm fixed, and thus, UP-CLUC cannot control the tradeoff adaptively according to the context arrivals. On the other hand, when AP-CLUC learns, it optimizes its partition to address the tradeoff in an online manner by partitioning the context space on-the-fly according to the context arrivals. In AP-CLUC, for the sets in regions of the context space with a large number of context arrivals, partitioning the sets smaller is favorable for reducing the total error, since the approximation error is reduced due to the smaller sets and the estimation error will become small soon due to the large number of context arrivals. Thus, as shown in Fig. 2, AP-CLUC partitions such regions into smaller sets. This results in a partition where smaller sets are concentrated around the regions of the context space with a large number of context arrivals and larger sets are scattered over the remaining parts of the context space with a small number of context arrivals. In the literature, this phenomenon is called *contextual zooming* [41]. Owing to such an optimization, AP-CLUC outperforms UP-CLUC regardless of the system parameter of UP-CLUC as will be shown in the numerical results section.

For more effective learning, we can generate virtual experiences assumed that the unselected actions were selected owing to the nature of the UC problem. By using such virtual experiences, we can accelerate the learning speed of AP-CLUC. In addition, when AP-CLUC partitions a region into smaller sets, the costs due to learning the uncertainties in the smaller sets newly can be reduced by reusing the past experiences learned in the region. The details of such methods for effective learning will be described in the following subsection, and their effects will be shown in the numerical results section.

B. Algorithm Description of AP-CLUC

The pseudocode of AP-CLUC is given in Algorithm 1. For simplicity of the description, we assume that $T_{sc} = 0$ and normalize the extended context[2] space to be $\mathcal{Z} = [0,1]^D$, where D is the dimension of the context space, i.e., $D_{\mathcal{X}} + J$. It is worth noting that normalizing the context is used only for the performance analyses, and the performance bound of AP-CLUC can always be achieved by a proper scaling of the context. Also, the assumption does not affect the performance bound. How the partitioning of the context space in AP-CLUC is different from that in UP-CLUC is illustrated in Fig. 2 for $D = 2$. Unlike UP-CLUC, i.e., Algorithm 1 in [1], which uniformly partitions the context space at the beginning of the algorithm, in AP-CLUC the context space is partitioned into smaller sets on-the-fly according to the context arrivals. This adaptive partitioning enables AP-CLUC to learn more precisely on the frequent context information. In AP-CLUC, every partition of the context space is composed of hypercubes with side lengths belonging to the set $\{2^0, 2^{-1}, 2^{-2}, ...\}$, and a D-dimensional hypercube which has sides of length 2^{-l} is called a level l hypercube. The partition at a time period is composed of a set of disjoint hypercubes that cover the context space. This set of hypercubes are also referred to as the *active* hypercubes at that time period. The set of active hypercubes, i.e., the context partition, is denoted by $\mathcal{P}_{\mathcal{Z}}$.

The adaptive partitioning mechanism of AP-CLUC performs as follows. The initial context partition for AP-CLUC is given by $\mathcal{P}_{\mathcal{Z}} = \{[0,1]^D\}$ as given in line 1 of Algorithm 1, which is the entire context space (i.e., $l = 0$). Then, according to the context arrivals, this partition is updated by the mechanism described below. Let $l(p_z)$ be the level of hypercube p_z and $N(p_z)$ be the number of context arrivals to hypercube p_z after p_z was activated. An active hypercube p_z is deactivated if $N(p_z) \geq 2^{\rho l(p_z)}$ as in line 13 of Algorithm 1, where $\rho > 0$ is a parameter of AP-CLUC. When p_z is deactivated, 2^D level $l(p_z)+1$ child hypercubes formed by partitioning hypercube p_z become active, and the context partition is updated as $\mathcal{P}_{\mathcal{Z}} \cup G_{p_z}^{l(p_z)+1} \setminus \{p_z\}$ as in line 16 of Algorithm 1, where $G_{p_z}^{l(p_z)+1}$ is the set of 2^D level $l(p_z)+1$ child hypercubes created from the hypercube p_z. This adaptive partitioning is illustrated in Fig. 2 for $D = 2$ and $\rho = 1$.

[2]In algorithm description, we omit "extended" from the extended context for convenience.

In adaptive partitioning, the deactivation process of a hypercube depends on its level. In addition, the action space is also adaptively discretized according to the level of the hypercube that the context belongs to. To this end, the slicing parameter for the power output which is used to discretize the power output is determined by the level of the hypercube l, and hence, is denoted by $m_A(l)$. Then, the power output of unit j is uniformly discretized using $m_A(l)$. The set of the discretized power outputs of unit j for a level l hypercube is denoted by $\bar{P}_j(l) = \{p_j^{\min} + p_j^{m_A(l)}, p_j^{\min} + 2p_j^{m_A(l)}, ..., p_j^{\max}\}$, where $p_j^{m_A(l)} = (p_j^{\max} - p_j^{\min})/m_A(l)$. The power output of unit j during time period t is denoted by $\bar{p}_j(t) \in \bar{P}_j(l(p_z(\mathbf{z}(t))))$, where $l(p_z(\mathbf{z}))$ represents the level of hypercube p where context $\mathbf{z}$ belongs to. The vector of the discretized power outputs of all units during time period t is denoted by $\mathbf{p}_{ther}(t) = \{\bar{p}_j(t)\}_{\forall j \in \mathcal{J}}$ and the discretized action space for a level l hypercube is given by $\bar{\mathcal{A}}(l) = \{0,1\}^J \times \prod_{j \in \mathcal{J}} \bar{P}_j(l)$. Using these, the set of discretized available actions for unit status $\mathbf{s}$ and a level l hypercube is given as

$$\bar{\mathcal{A}}(\mathbf{s}, l) := \{\{\mathbf{u}(t), \bar{\mathbf{p}}_{ther}(t)\} |\ (1),\ (2), (3) \text{ and } (4) \text{ holds}\}.$$

Remark 2: Note that in the early stages of running AP-CLUC, there might exist no available action satisfying the ramp rate limit constraint in (3) when the discretization of power outputs is too coarse, i.e., there are only a few discretized power outputs. This problem can be resolved by appropriately setting $m_A(l)$ of low levels to be large enough to satisfy the ramp rate limit constraint.

We denote the number of times that action a is chosen when the context is in active hypercube p_z as $N(a, p_z)$. We also define the estimated cost of action a on set p_z, $\hat{c}(a, p_z)$, which represents the sample mean of the total operating cost observed from action a on active hypercube p_z. At the beginning of each time period t, the system observes the context $\mathbf{z}(t)$ and unit status $\mathbf{s}(t)$. Then, it finds the corresponding active hypercube $p_z(\mathbf{z}(t))$ that the current context belongs to and calculates the set of available actions $\bar{\mathcal{A}}(\mathbf{s}(t), l(p_z(\mathbf{z}(t))))$ given the unit status. Then, it chooses the action with the lowest estimated total operating cost given as

$$\hat{\pi}(t) \in \underset{a' \in \bar{\mathcal{A}}(\mathbf{s}(t), l(p_z(\mathbf{z}(t))))}{\operatorname{argmin}} \hat{c}(a', p_z(\mathbf{z}(t))).$$

During the time period t, the system operates its thermal power generation units according to the chosen action $\hat{\pi}(t)$. At the end of the time period, the system observes the realization of the uncertainties with which the total operating cost during the time period, $C_{tot}(t)$, is obtained as in (5). Then, the system updates the estimated cost $\hat{c}(\hat{\pi}(t), p_z)$ by using $C_{tot}(t)$ in line 8 of Algorithm 1.

The selected action does not affect the distribution of the uncertain events that happen in the current time period. This nature of the UC problem allows us to calculate the total operating cost for the actions that are not selected, i.e., $a \in \bar{\mathcal{A}}(l(p_z)) \setminus \{\hat{\pi}(t)\}$, from the observed cost. Note that for each unselected action, the fuel cost in (6) and the start-up cost in (7) can be simply calculated. The load shedding cost in (8) and power curtailment cost in (9) can be also calculated by using the realized load demand and renewable power outputs due to the nature of the UC problem. Thus, by using the calculated costs for the unselected actions, i.e., *virtually* observed costs, AP-CLUC performs *virtual* updates of the estimated costs of unselected actions in order to accelerate the learning as given in lines 10–11 of Algorithm 1. Therefore, the number of times that action a is (virtually) chosen when the context is in the set p_z, i.e., $N(a, p_z)$, is updated for all $a \in \bar{\mathcal{A}}(l(p_z))$.

Algorithm 1: AP-CLUC.

1: $\mathcal{P} = \{[0,1]^D\}$,
2: $\hat{c}(a, [0,1]^D) \leftarrow \infty$ and $N(a, [0,1]^D) \leftarrow 0, \forall a \in \bar{\mathcal{A}}(0)$
3: $N([0,1]^D) \leftarrow 0$
4: **while** TRUE **do**
5: Observe context $\mathbf{z}$ and unit status $\mathbf{s}$
6: $p \leftarrow p_z(\mathbf{z})$, $a \leftarrow \operatorname{argmin}_{a' \in \bar{\mathcal{A}}(\mathbf{s}, l(p))} \hat{c}(a', p)$
7: Operate units with a and observe C_{tot}
8: $\hat{c}(a,p) \leftarrow \frac{\hat{c}(a,p)N(a,p)+C_{tot}}{N(a,p)+1}$
9: *Virtually* observe $C'_{tot}(a'), \forall a' \in \bar{\mathcal{A}}(l(p)) \setminus \{a\}$
10: $\hat{c}(a',p) \leftarrow \frac{\hat{c}(a',p)N(a,p)+C'_{tot}(a')}{N(a,p)+1}, \forall a' \in \bar{\mathcal{A}}(l(p)) \setminus \{a\}$
11: $N(a,p) \leftarrow N(a,p) + 1,\ \forall a \in \bar{\mathcal{A}}(l(p))$
12: $N(p) \leftarrow N(p) + 1$
13: **if** $N(p) \geq 2^{\rho l(p)}$ **then**
14: Create 2^D level $l(p)+1$ child hypercubes, $G_p^{l(p)+1}$
15: Run INIT $\left(G_p^{l(p)+1}, p\right)$
16: $\mathcal{P} \leftarrow \mathcal{P} \cup G_p^{l(p)+1} \setminus \{p\}$
17: **end if**
18: **end while**
19: **procedure** INIT(B, p)
20: **for** $p' \in \mathcal{B}$ **do**
21: $\hat{c}(a,p') \leftarrow \infty$ and $N(a,p') \leftarrow 0, \forall a \in \bar{\mathcal{A}}(l(p'))$
22: $N(p') \leftarrow 0$
23: **end for**
24: **end procedure**

Remark 3: Note that the virtual update allows AP-CLUC to accelerate the learning speed, but it also causes an increase in the computational complexity of AP-CLUC due to the calculation of the costs for the unselected actions. We can control the tradeoff between the learning speed and the computational complexity by performing the virtual updates for a part of the unselected actions, not for all unselected actions. We investigate the computational complexity according to the number of the virtually updated unselected actions in Section IV-D, and show the learning speed in the numerical results.

Moreover, to help AP-CLUC learn faster, when child hypercubes become active, we can reuse the a priori information provided from their parent hypercube. To this end, we initiate the parameters of activated hypercubes using Algorithm 2 instead of using the initiating procedure in AP-CLUC as given in lines 21–22 of Algorithm 1. We call this an *experience reuse*, and it helps AP-CLUC learn faster by providing a guidance in the early stages of learning in the activated hypercubes. This improvement is shown in the numerical results section.

Algorithm 2: Experience Reuse.

procedure ExpReuse(B, p)
  for $p' \in \mathcal{B}$ **do**
    $N(a, p') \leftarrow \lceil N(a,p)/2 \rceil, \forall a \in \bar{\mathcal{A}}(l(p'))$
    $N(p') \leftarrow \lceil N(p)/2 \rceil$
    $\hat{c}(a, p') \leftarrow \hat{c}(\tilde{a}, p), \forall a \in \bar{\mathcal{A}}(l(p'))$, where $\tilde{a}$ is the action in $\bar{\mathcal{A}}(l(p))$ which is nearest from a.
  end for
end procedure

C. Regret Bound for AP-CLUC

In this subsection, to evaluate the performance of AP-CLUC in Algorithm 1, we first define the learning regret, and then provide the regret bound for AP-CLUC. For simplicity of the presentation, we normalize the total operating cost such that it lies in $[0, 1]$. Let the expected operating cost of action $a \in \mathcal{A}$ during a time period with a given context $\mathbf{z} \in \mathcal{Z}$ be

$$c(a, \mathbf{z}) := \mathbb{E}_{\hat{M}(\mathbf{x}), \hat{p}_{re}(\mathbf{x})} \left[C_{tot}(a, t) \right],$$

where $\hat{M}(\mathbf{x})$ and $\hat{p}_{re}(\mathbf{x})$ are the random variables for the load demand and the renewable power output, respectively, during the time period where the context $\mathbf{x}$ is given. The joint distribution of $\hat{M}(\mathbf{x})$ and $\hat{p}_{re}(\mathbf{x})$ is given by $F_{\mathbf{x}}$. Next, we show that the expected total operating costs are similar for similar contexts, which is widely used as a similarity information [42], [43]. We formalize this as a Lipschitz condition, and we prove that the expected cost of each action also satisfies the Lipschitz condition in the following lemma.

Lemma 1: There exists $L > 0$ such that for all $\mathbf{z}$, $\mathbf{z}' \in \mathcal{Z}$ and $a \in \mathcal{A}$,

$$|c(a, \mathbf{z}) - c(a, \mathbf{z}')| \leq L \|\mathbf{z} - \mathbf{z}'\|,$$

and for all a, $a' \in \mathcal{A}$ and $\mathbf{z} \in \mathcal{Z}$, where the on/off status are same but the power outputs might be different,

$$|c(a, \mathbf{z}) - c(a', \mathbf{z})| \leq L \|a - a'\|.$$

Proof: See Appendix A. ■

We define the regret with respect to a complete information benchmark, which myopically selects the best available action for the current time period given perfect knowledge of the distribution $F_{\mathbf{x}}$, i.e., the impact on the future costs is not considered when selecting the action. It is worth noting that in a viewpoint of the existing UC approaches, $F_{\mathbf{x}}$ can be interpreted as a target distribution which their a priori information is intended to provide. Given context $\mathbf{z}$ and unit status $\mathbf{s}$, this benchmark is defined as

$$\pi^*_{\mathbf{s}}(\mathbf{z}) := \operatorname{argmin}_{a \in \mathcal{A}(\mathbf{s})} c(a, \mathbf{z}), \forall \mathbf{z} \in \mathcal{Z}. \tag{11}$$

It is worth emphasizing that the complete information benchmark is defined on the continuous action space $\mathcal{A}$. Let $\hat{\pi}$ be the UC policy obtained by AP-CLUC. Then, the expected learning regret with respect to the benchmark $\pi^*_{\mathbf{s}}(\mathbf{z})$ in (11) by time period T is given by

$$R(T) := \mathbb{E}\left[\sum_{t=0}^{T} C_{tot}(\hat{\pi}(t), t) - \sum_{t=0}^{T} c(\pi^*_{\mathbf{s}(t)}(\mathbf{z}(t)), \mathbf{z}(t)) \right] \tag{12}$$

where $\mathbf{s}(t)$ and $\mathbf{z}(t)$ denote the unit status and context of AP-CLUC at time period t.

The following theorem bounds the regret of AP-CLUC (without experience reuse) given in (12).

Theorem 1: When the parameters of AP-CLUC are set as $\rho = 2(J+1)$ and $m_A(l) = 2^l$, we have for AP-CLUC

$$R(T) \leq \sum_{l=1}^{\lceil \frac{\log_2 T}{2(J+1)} \rceil + 1} K_l(T) \left[2^{l(2J+1)} (2L\sqrt{D} + 2^{J+\frac{1}{2}} + L\bar{p}_{ther}\sqrt{J}) + 1 \right],$$

where $K_l(T)$ is the number of level l hypercubes that are activated by time T and $\bar{p}_{ther} = \max_{j \in \mathcal{J}} [p_j^{\max} - p_j^{\min}]$.

Proof: See Appendix B. ■

Note that the regret bound for AP-CLUC depends the number of activated level l hypercubes given by $K_l(T)$, which depends on the pattern of context arrivals. Using the general form of the regret bound given in Theorem 1, next we show that the regret of AP-CLUC for the worst possible pattern of context arrivals which maximizes the number of activated hypercubes (in which the contexts arrive uniformly over the context space) is sublinear in T.

Corollary 1: When the parameters of AP-CLUC are set as $\rho = 2(J+1)$ and $m_A(l) = 2^l$, if the context arrivals by time T are uniformly distributed over the context space, we have for AP-CLUC,

$$R(T) = O\left(T^{\frac{D+2J+1}{D+2J+2}} \right).$$

Proof: See Appendix C. ■

The regret bound in Collorary 1 is sublinear in T. Thus, in theory, with the indefinite time periods, it is guaranteed that the average cost of AP-CLUC converges to the average cost of the benchmark, i.e., $\lim_{T \to \infty} R(T)/T = 0$, for all possible context arrivals.

D. Computational Complexity of AP-CLUC

In each time period, AP-CLUC needs to perform comparison operations to identify the active hypercube that the current context belongs to. The computational complexity of such identification is given by $O(|\mathcal{P}|)$, where $|\mathcal{P}|$ is the cardinality of the active hypercubes in the partition $\mathcal{P}$. Since the uniform context arrivals over the context space maximizes the number of active hypercubes, $|\mathcal{P}|$ at time period T is bounded by $|\mathcal{P}| < 2^{D l_{\max}(T)}$, where $l_{\max}(T) = \lfloor 1 + \frac{\log_2 T}{D+\rho} \rfloor$ is the maximum hypercube level at time period T with the uniform context arrivals derived in Appendix C. Then, the worst-case computational complexity of the identification at time period T is given by $O(2^{D l_{\max}(T)})$.

After identifying the active hypercube, AP-CLUC needs to perform one comparison operation for choosing the action with

the lowest estimated total operating cost and update operations on the estimated total operating costs of the actions including the unselected actions, i.e., virtual updates. To virtually update the estimated total operating costs of the unselected actions, the operating costs of the unselected actions have to be computed from the observed cost of the selected action and realization of the uncertainties. Thus, the computational complexity of the virtual updates highly depends on the number of the unselected actions whose estimated total operating costs will be virtually updated in each time period. The computational complexity of the update operations has the order $O((2m_A(l))^J)$ if the estimated costs for all unselected actions are virtually updated. On the other hand, if AP-CLUC does not virtually update any unselected actions, then the computational complexity has the order $O(1)$. Hence, we can control the computational complexity of AP-CLUC by limiting the number of the unselected actions whose estimated total operating costs will be virtually updated.

V. Numerical Results

In this section, we provide simulation results to evaluate the performance of AP-CLUC.

A. Simulation Setup

The length of a time period is taken to be an hour and the circular time duration for the context is set to be a day, i.e., $\mathcal{H} = \{0, 1, ..., 23\}$. It is worth emphasizing that AP-CLUC starts without any a priori information of the uncertainties and learns them during the simulation. We consider a microgrid with wind turbines and four identical thermal power generation units. The parameters of the thermal power generation units are provided in Table II. We set the load shedding price, LSP, and the power curtailment price, PCP, to be 200 \$/kWh [9]. The spinning reserve requirement is set to be 10% of the total power output of the thermal power generation units.

In our simulation, we consider a context consisting of current time, load demand context, weather context, and down time of units, where the dimensions of both load demand and weather context spaces are 1.[3] The power output profile for each hour and parameters of wind turbines are adopted from [9], and their power output capacity is set to be 650 kW. For a load demand profile for each hour, we use the hourly average load shapes of residential electricity services in California [44] with 500 customers. Then, the uncertainties are generated by using their profiles and the context. In each time period t, the load demand context, $x_M(t)$, is non-uniformly generated between $[-1, 1]$ by a truncated normal distribution whose mean is zero and variance is 0.2. Then, the load demand is generated by a Gaussian distribution of which mean is set to be $P_M^{profile}(h(t)) + x_M(t)P_M^{uncert}$, where $P_M^{profile}(h)$ is the value of the load demand profile in time index h and P_M^{uncert} is an amount of load demand uncertainty. Note that the Gaussian distribution is widely used to model the forecasting error [45]. By this, the load demand is generated based on both its profiles and its correlated context. The standard deviation of the distribution is set to be 2.5% of its mean, which is widely used to model the scenarios in day-ahead UC problems [9], [46]. Similarly, the weather context, $x_W(t)$, is non-uniformly generated between $[-1, 1]$ by the same distribution for the load demand context. Then, the renewable power output is also generated by a Gaussian distribution of which mean is set to be $P_W^{profile}(h(t)) + x_W(t)P_W^{uncert}$, where $P_W^{profile}(h)$ is the value of the renewable power output profile in time index h and P_W^{uncert} is an amount of renewable power uncertainty. We set both P_M^{uncert} and P_W^{uncert} to be 150 kW.

TABLE II
Parameters of Thermal Units [9]

p_j^{max}(kW)	p_j^{min}(kW)	$C_{j,fu}^{(1)}$(\$/Hr)	$C_{j,fu}^{(2)}$(\$/kWh)	$C_{j,fu}^{(3)}$(\$/kW^{2}h)
100	50	55	17	0.0012
RR_j(kW)	MUT_j(Hr)	MDT_j(Hr)	CST_j(Hr)	CSC_j(\$)
100[4]	3	3	0	1,140

To evaluate the performance of AP-CLUC, we compare it with its complete information benchmark and also with the stochastic optimization for UC (SOUC) that is one of the most representative UC approaches [2]–[9], [28]. For SOUC, we assume that *perfect* a priori information (PI) is given. We also compare it with several learning algorithms: AP-CLUC without experience reuse (ER), UP-CLUC, Q-learning-like algorithm for UC (QLUC), UCB1, and EXP3. The descriptions of the algorithms and their parameter settings are provided as follows. The number of power outputs, m_A, is set to be 8 for all algorithms unless mentioned explicitly.

- SOUC with PI chooses the best UC schedule considering all possible stochastic scenarios on the uncertainties using the perfect information. In other words, it chooses the optimal UC schedule in the continuous action space $\mathcal{A}$ as in (11). It is worth noting that it is an ideal SOUC since such perfect a priori information cannot be obtained in reality.
- AP-CLUC is given in Algorithm 1 with ρ which is set to be 2. In AP-CLUC, we also run the complete information benchmark. The benchmark chooses the optimal UC schedule as in (11). This is similar to SOUC with PI, but when the benchmark chooses the UC schedule, it uses the unit status and context of AP-CLUC as in (12), while SOUC with PI uses its own. We also implement AP-CLUC without ER to show the improvement due to ER. In AP-CLUC without ER, for initiating of each hypercube, its initial estimated cost is set to be infinite and its counters are set to be zero.
- UP-CLUC is given in [1], and we implement two UP-CLUCs with m_z set to be 5 and 10, respectively, where m_Z is the slicing parameter for the context space.
- QLUC is a learning algorithm which consider only the current time information which is basic state information while not considering both load demand and weather contexts. We simply implement it by neglecting both contexts

[3]To simply construct the simulation system with stochastic uncertainties, we assume that the dimension of each of the load demand and weather contexts is one. In real world, the projected context can be obtained as discussed in Remark 1.

[4]Note that the ramp rates of the units are assumed to be equal to $p_j^{\max}$ since their sizes are small enough to reach their maximum power outputs within a time period [9].

TABLE III
ON/OFF STATUS OF UNITS BY AP-CLUC

Unit no.	On/off status of units (ON=1, OFF=0)																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
2	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0
3	1	1	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

TABLE IV
COMPARISON OF AVERAGE COSTS ($)

	C_{fuel}	C_{su}	C_{sh}	C_{cu}	C_{tot}
SOUC with PI	2,524	2,460	3,351	2,983	11,317
AP-CLUC	2,455	2,347	4,868	3,722	13,391
Benchmark	2,510	2,444	3,375	2,853	11,182
UP-CLUC ($m_z=10$)	2,494	2,382	4,779	4,076	13,731
UP-CLUC ($m_z=5$)	2,459	2,331	5,770	4,712	15,272
QLUC	2,205	2,196	8,909	5,141	18,450
UCB1	1,364	1,413	16,870	2,697	22,344
EXP3	2,492	2,427	11,260	9,864	26,043

in UP-CLUC. In QLUC, we also adopt the virtual updates for fair comparison.

The following learning algorithms, i.e., UCB1, and EXP3, do not consider the context information. The parameters of each algorithm are chosen as the set of parameters for which the algorithm performs the best.

- UCB1 [47] computes an index for each action, which is a lower confidence bound of the expected cost. Then, the algorithm chooses the action with has the lowest index.
- EXP3 [48] computes and updates a weight parameter for each action by using its realized operating costs. Then, it uses the weight parameters to randomly decide the action to be taken. For EXP3, m_A is set to be 4 instead of 8 for better performance.

To clearly show that AP-CLUC addresses the UC problem, we list the on/off status of units by AP-CLUC for certain 24 time periods during the simulation in Table III. According to the observed context that is strongly correlated with the uncertainties, AP-CLUC decides the on/off status of units to minimize the average total operating costs. From the table, we can see that AP-CLUC satisfies the minimum up/down time constraints.

B. Average Costs and Learning Speeds

We first compare the achieved average costs, which are provided in Table IV. CLUCs, i.e., AP-CLUC and UP-CLUCs, achieve better performance than other learning algorithms which do not utilize the context information. Especially, AP-CLUC achieves 27.4%, 40.1%, and 48.6% cost reduction against QLUC, UCB1, and EXP3, respectively. This result shows that using the context information is effective to achieve better performance when the system uncertainties are correlated to the context information. It is worth noting that many existing researches show that the context information, i.e., current time, weather condition, and past load demand, is highly correlated to the system uncertainties, i.e., renewable power outputs and load demand, in real world [33], [34]. In addition, QLUC which uses only the current time context achieves better performance

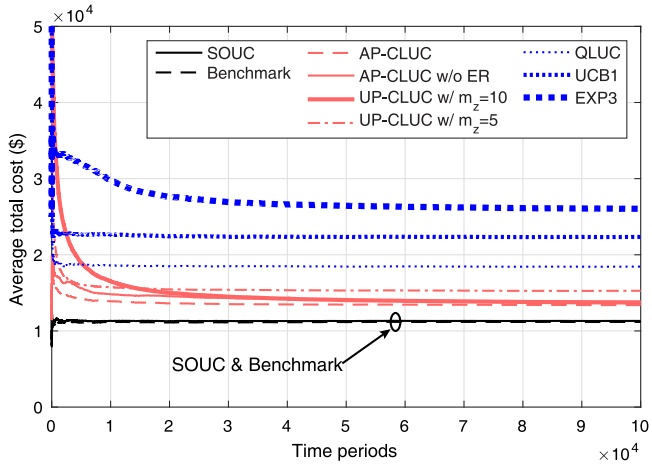

Fig. 3. Average total operating costs of the algorithms.

than other learning algorithms which do not use any context information. This result also shows the effectiveness of using the context information. We can see that in general the load shedding costs of the algorithms which do not use the context information are larger than those of CLUCs, while their fuel costs are smaller. This result shows that in general they generate too small amount of power to support the load demand compared with CLUCs since they fail to predict the uncertainties due to the lack of the context information.

In addition, from Table IV, we see that AP-CLUC achieves a performance close to the benchmark by effectively using the context information. It is worth noting that the benchmark is based on the assumption that it has perfectly accurate a priori information. The benchmark and SOUC with PI achieve a similar performance owing to the perfect information. AP-CLUC and UP-CLUC which has a fine partition of the context space, i.e., UP-CLUC with $m_z = 10$, achieve performance close to SOUC with PI by effectively using the context information. On the other hand, UP-CLUC which has a rough partition of the context space, i.e., UP-CLUC with $m_z = 5$, only achieves worse performance than them due to the approximation errors from merging context arrivals.

In Fig. 3, we compare the learning speeds of the learning algorithms. The faster learning speed of a learning algorithm implies that when the statistical characteristics of the system uncertainties vary, the algorithm can adapt to it more quickly. Hence, the learning speed of the learning algorithm is important to use it in practice, since in real-world, such statistical characteristics might vary over time due to many environmental reasons such as seasonal change and economy. Note that SOUC and benchmark are not learning algorithms. We can see that AP-CLUC,

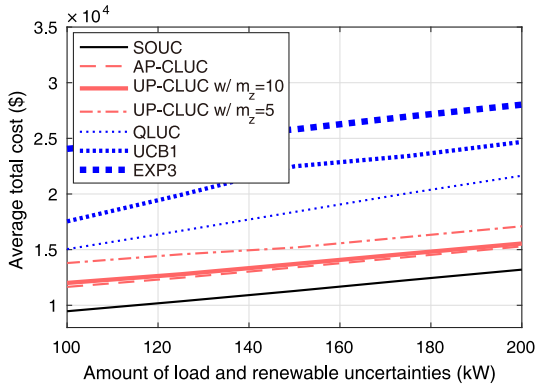

Fig. 4. Average total operating costs of the algorithms varying the amount of uncertainties.

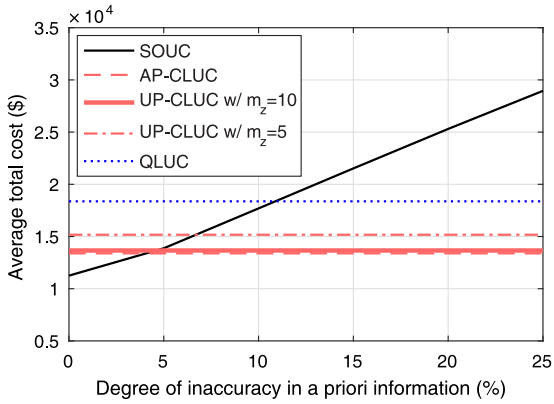

Fig. 5. Average total operating costs of CLUCs, QLUC and SOUC with inaccurate a priori information.

AP-CLUC without ER, UP-CLUC with $m_z = 5$, QLUC, and UCB1 have relatively fast learning speeds, and UP-CLUC with $m_z = 10$ and EXP3 have relatively slow learning speeds. By comparing UP-CLUCs, we can see that UP-CLUC which has a finer partition of the context space learns slower while achieving smaller average cost. In addition, by reusing the past experience, AP-CLUC learns faster than AP-CLUC without ER as shown in the figure.

C. Impact of Degree of Uncertainties

We see the impact of uncertainties by varying the degree of uncertainties in our simulation system, i.e., P_M^{uncert} and P_W^{uncert}. Note that the degree of uncertainty represents the maximum deviation from the profile value according to the context information. Thus, as the amount of uncertainties increases, both load demand and renewable power output more fluctuate. In general, more fluctuation of the uncertainties causes higher operating costs since the system needs more effort to address the uncertainties. From Fig. 4, we can see that the average total operating costs of all learning algorithms increase as the degree of uncertainties increases. It is worth noting that the average cost of SOUC with PI also increases even it has perfect a priori information since more inevitable costs occur from the constraints of the thermal units such as the minimum power outputs of the units and the minimum up/down times. In the figure, we can also see that the increased amounts of the average total costs of CLUCs are similar with that of SOUC with PI. This shows that CLUCs can address the uncertainties only incurring a similar amount of cost when using SOUC with PI since they can learn the uncertainties by using the context information. On the other hand, the increased amount of the average total costs of the learning algorithms which do not use the load demand and weather contexts is larger than that of other algorithms which use the context information. This also implies that using the context information is effective to address the uncertainties.

D. Impact of Inaccuracy of a Priori Information

Next, we evaluate the performance of SOUC when the a priori information is inaccurate. Specifically, we investigate the inaccuracy of a priori information on the performance of SOUC. It is worth noting that perfectly accurate a priori information cannot be obtained in reality, and thus, any a priori information used in the existing UC models is inaccurate to some degree. To adjust the degree of inaccuracy in a priori information, we consider the case when the mean of the load demand is overestimated and the mean of the renewable power output is underestimated compared to their expected values. The degree of overestimation and underestimation is stated in percentages. In Fig. 5, we show the average total operating costs of CLUCs, QLUC, and SOUC varying the degree of inaccuracy in a priori information. For simple presentation, among the comparative learning algorithms, i.e., QLUC, UCB1 and EXP3, only the performance of QLUC is provided in the figure since QLUC has the best performance among them. We can see that the average total cost of SOUC increases as the degree of inaccuracy increases. On the other hand, the average total cost of other algorithms does not change since they do not use a priori information. AP-CLUC and UP-CLUC with $m_z = 10$ achieve better performance than SOUC if the degree of inaccuracy in a priori information of SOUC becomes more than 4%. Moreover, the difference between the performances of CLUCs and SOUC rapidly increases as the degree of inaccuracy increases, and if the degree of inaccuracy becomes more than 11%, SOUC has worse performance than even QLUC which uses only the current time context. This result shows that CLUCs are more effective than SOUC when given a priori information is not highly accurate.

E. Effectiveness of Adaptive Partitioning

In this subsection, we compare AP-CLUC and UP-CLUC to show the effectiveness of adaptive partitioning in AP-CLUC. From the previous results, we can infer that in UP-CLUC, there is a tradeoff between the average costs and the learning speed and the tradeoff can be controlled by the slicing parameter for the context space m_z. To compare the performance of AP-CLUC and UP-CLUC more clearly, in Fig. 6, we provide the average total operating costs of AP-CLUC and UP-CLUC varying m_z as 2, 4, 6, and 10. From the figure, we can see that in UP-CLUC, as m_z increases, the learning speed becomes slower, but the average total cost decreases with enough time periods. Thus, due to such slow learning speeds, UP-CLUC with large m_z

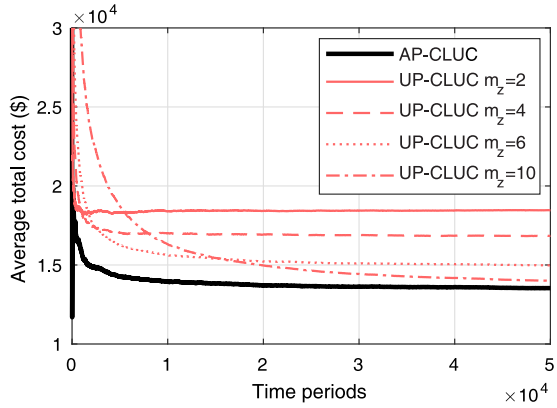

Fig. 6. Average total operating costs of AP-CLUC and UP-CLUC varying m_z as 2, 4, 6, and 10.

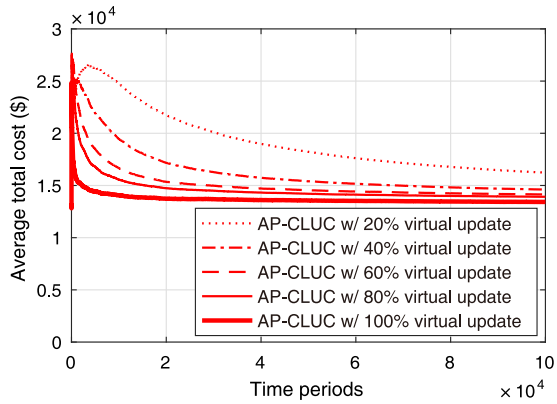

Fig. 7. Average total operating costs of AP-CLUCs varying the ratio of virtually updated unselected actions from all unselected actions.

has a worse average total cost than UP-CLUC with smaller m_z before the uncertainties are learned enough. On the other hand, AP-CLUC does not have such a tradeoff since it adaptively partitions the context space according to the context arrivals. In the figure, AP-CLUC achieves the lowest average total cost while having a relatively fast learning speed compared with UP-CLUCs. Besides, it achieves the lower average total cost than UP-CLUC in all time periods. regardless of m_z. This implies that AP-CLUC outperforms UP-CLUC regardless of m_z owing to its adaptive partitioning.

F. Impact of Virtually Updated Actions

In Fig. 7, the impact of the number of virtually updated unselected actions in AP-CLUC is shown. The unselected actions which will be virtually updated are randomly chosen, and their numbers are determined as 20%, 40%, 60%, 80%, and 100% of all unselected actions. We can see that as more number of unselected actions are virtually updated, the learning speed of AP-CLUC increases. However, as investigated in Section IV-D, the computational complexity also increases. Thus, the number of virtually updated unselected actions should be carefully chosen considering the tradeoff between the learning speed and the computational complexity.

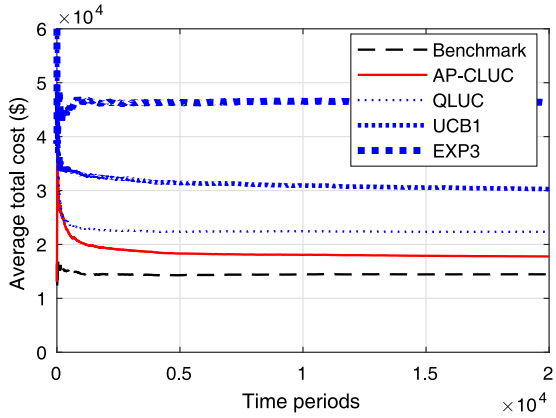

Fig. 8. Average total operating costs of the algorithms in the larger microgrid.

G. Average Total Operating Costs and Learning Speeds in Larger Microgrids

In Fig. 8, we provide the average total operating costs of the algorithms in a microgrid having a larger number of customers, wind turbines, and thermal power generation units compared with the microgrid considered in the previous results. In the microgrid, we consider six units, 750 customers, and wind turbines whose power output capacity is given by 1000 kW. Similar to the previous results, AP-CLUC achieves better performance than other learning algorithms and a performance close to the benchmark. Moreover, the learning speed of AP-CLUC is similar to that in the previous results. This clearly shows that AP-CLUC is applicable to larger microgrids.

VI. Conclusion and Future Work

In this paper, we developed AP-CLUC which minimizes the average total operating cost of the microgrid with renewable energy sources by learning the system uncertainties using the context information. Then, we proved the optimality of AP-CLUC in terms of the long-term average cost. Moreover, we showed through simulations that AP-CLUC achieves performance close to the complete information benchmark which has perfect a priori information about the system uncertainties, and outperforms other learning algorithms which do not use the context information. Our results show that two key properties of AP-CLUC, use of and adaptive management of the context information, makes it perform better than its competitors including UP-CLUC.

As a future work on this subject, power flow issues can be incorporated into the UC problem for the secure power flow. To this end, the transmission line capacity constraints can be considered in AP-CLUC. Moreover, AP-CLUC can be extended by incorporating power flow decisions into the actions. In addition to the power flow issues, the operational reliability of microgrids can be also considered in AP-CLUC. For example, the existing concepts to adjust the conservativeness and robustness, such as minimax regret [14] and CVaR [15], can be applied to AP-CLUC. Moreover, the reliability for load shedding or power curtailment can be addressed by incorporating more strict reserve requirement constraints or introducing the different weights for each type of cost. Lastly, our learning

approach can be extended to UC scenarios on microgrids using game theory, which are widely studied recently [49]–[51]. For this, there are several promising learning methods such as game-theoretical multi-armed bandits [52], which can be used for such scenarios with game settings.

APPENDIX A
PROOF OF LEMMA 1

The proof is done by showing that all costs including the operating cost in (5), the fuel cost in (6), the start-up cost in (7), the load shedding cost in (8) and the power curtailment cost in (9) obeys to the Lipschitz condition for context $\mathbf{z}$ and action a. We assume that the statistical characteristics of load demand and renewable power outputs are similar for similar contexts, and formalize this as the Lipschitz condition. For the simple presentation, we substitute $\hat{M}(\mathbf{x}) - \hat{p}_{re}(\mathbf{x})$ as $\omega_{\mathbf{x}}$ and denote an event $\{a < \omega_{\mathbf{x}} \leq b\}$ by $\Omega_a^b(\mathbf{x})$.

Assumption 1: There exists $L_x > 0$ such that for all $\mathbf{x}$, $\mathbf{x}' \in \mathcal{X}$, we have

$$\left|\mathbb{E}[\omega_{\mathbf{x}}\mathbb{I}(\Omega_a^b(\mathbf{x}))] - \mathbb{E}[\omega_{\mathbf{x}'}\mathbb{I}(\Omega_a^b(\mathbf{x}'))]\right| \leq L_x\|\mathbf{x}-\mathbf{x}'\|$$

and

$$|\mathbf{P}(\Omega_a^b(\mathbf{x})) - \mathbf{P}(\Omega_a^b(\mathbf{x}'))| \leq L_x\|\mathbf{x}-\mathbf{x}'\|$$

for any given $a \leq b$, where $\mathbb{I}(\Omega)$ is the indicator function for event Ω and $\mathbf{P}(\Omega)$ denotes the probability of event Ω.

The proof for $\mathbf{z}$*:* The fuel cost always satisfies the condition since it does not depend on the extended context $\mathbf{z}$. To prove that the start-up cost of unit j satisfies the condition, we show that

$$CSC_j\left|e^{-\frac{T_{j,off}}{CST_j}} - e^{-\frac{T'_{j,off}}{CST_j}}\right| \leq L\left|T_{j,off} - T'_{j,off}\right|, \quad (13)$$

for some $L > 0$, where $T_{j,off}$ and $T'_{j,off}$ are the elements representing the down times of unit j in $\mathbf{z}$ and $\mathbf{z}'$, respectively. Since $T_{j,off} \geq 0$ and $T'_{j,off} \geq 0$, we have

$$\left|e^{-\frac{T_{j,off}}{CST_j}} - e^{-\frac{T'_{j,off}}{CST_j}}\right| \leq 1.$$

Using the fact that the down time is a non-negative integer, we get $\left\|T_{j,off} - T'_{j,off}\right\| \geq 1$, for any $T_{j,off}$ and $T'_{j,off}$ such that $T_{j,off} \neq T'_{j,off}$. Thus, when we choose $L = CSC_j$, the start-up cost of unit j satisfies the condition in (13) for any $T_{j,off}$ and $T'_{j,off}$ such that $T_{j,off} \neq T'_{j,off}$. When $T_{j,off} = T'_{j,off}$, the condition is satisfied regardless of L since both sides in (13) are zero. Hence, when we choose $L = CSC_j$, the start-up cost of unit j satisfies the condition for any $T_{j,off}$ and $T'_{j,off}$. From the load shedding cost in (8), we have

$$\begin{aligned}
&\left|LSP\left(\mathbb{E}([\omega_{\mathbf{x}} - \Sigma_j p_j]^+) - \mathbb{E}([\omega_{\mathbf{x}'} - \Sigma_j p_j]^+)\right)\right| \\
&= LSP\left|\mathbb{E}[(\omega_{\mathbf{x}} - \Sigma_j p_j)\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}))]\right. \\
&\qquad \left. -\mathbb{E}[(\omega_{\mathbf{x}'} - \Sigma_j p_j)\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}'))]\right| \\
&= LSP\left|\mathbb{E}[\omega_{\mathbf{x}}\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}))] - \mathbb{E}[\omega_{\mathbf{x}'}\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}'))]\right. \\
&\qquad \left. - \Sigma_j p_j\left(\mathbf{P}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x})) - \mathbf{P}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}'))\right)\right| \\
&\leq LSP\left|\mathbb{E}[\omega_{\mathbf{x}}\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}))] - \mathbb{E}[\omega_{\mathbf{x}'}\mathbb{I}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}'))]\right| \\
&\qquad + LSP\Sigma_j p_j\left|\mathbf{P}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x})) - \mathbf{P}(\Omega^\infty_{\Sigma_j p_j}(\mathbf{x}'))\right| \\
&\leq LSP \cdot L_x(1 + \Sigma_j p_j)\|\mathbf{x}-\mathbf{x}'\|,
\end{aligned}$$

where the last inequality follows Assumption 1. Then, by choosing $L = LSP \cdot L_x(1 + \Sigma_j p_j^{\max})$, we can show that the load shedding cost obeys to the Lipschitz condition. Similarly, we can show the Lipschitz condition of the power curtailment cost with $L = PCP \cdot L_x(1 + \Sigma_j p_j^{\max})$. Then, the expected cost is a Lipschitz continuous function of the extended context $\mathbf{z}$ with $L_{\mathbf{z}} = \sum_{j\in\mathcal{J}} CSC_j + L_x(LSP + PCP)(1 + \Sigma_j p_j^{\max})$.

The proof for a*:* To prove that the fuel cost of unit j satisfies the condition, we show that the following inequality is satisfied

$$c^{(1)}_{j,fu}(p_j - p'_j) + c^{(2)}_{j,fu}({p_j}^2 - {p'_j}^2) \leq L(p_j - p'_j),$$

for some $L > 0$, where $p_j > p'_j$. By dividing both sides of the above inequality by $(p_j - p'_j)$, we have

$$c^{(1)}_{j,fu} + c^{(2)}_{j,fu}(p_j + p'_j) \leq L.$$

Thus, when we choose $L = c^{(1)}_{j,fu} + 2c^{(2)}_{j,fu}p_j^{\max}$, the fuel cost of unit j satisfies the condition for any p_j and p'_j. The start-up cost satisfies the condition since it does not depend on the action. For the load shedding cost, we have

$$\left|LSP\left(\mathbb{E}([\omega_{\mathbf{x}} - \Sigma_j p_j]^+) - \mathbb{E}([\omega_{\mathbf{x}} - \Sigma_j p'_j]^+)\right)\right|.$$

To simplify the notations, we omit $\mathbf{x}$ in the following. Without loss of generality, we assume $\Sigma_j p_j > \Sigma_j p'_j$. Then, we can arrange the load shedding cost as

$$\begin{aligned}
&LSP\left|\mathbb{E}[(\omega - \Sigma_j p_j)\mathbb{I}(\Omega^\infty_{\Sigma_j p_j})] - \mathbb{E}[(\omega - \Sigma_j p'_j)\mathbb{I}(\Omega^\infty_{\Sigma_j p'_j})]\right| \\
&= LSP\left|\int_{\Omega^\infty_{\Sigma_j p_j}}(\omega - \Sigma_j p_j)dF - \int_{\Omega^\infty_{\Sigma_j p'_j}}(\omega - \Sigma_j p'_j)dF\right| \\
&= LSP\left|\int_{\Omega^\infty_{\Sigma_j p_j}}(\omega - \Sigma_j p_j)dF - \int_{\Omega^\infty_{\Sigma_j p_j}}(\omega - \Sigma_j p'_j)dF\right. \\
&\qquad \left. - \int_{\Omega^{\Sigma_j p_j}_{\Sigma_j p'_j}}(\omega - \Sigma_j p'_j)dF\right| \\
&= LSP\left|\int_{\Omega^\infty_{\Sigma_j p_j}}(\Sigma_j p_j - \Sigma_j p'_j)dF + \int_{\Omega^{\Sigma_j p_j}_{\Sigma_j p'_j}}(\omega - \Sigma_j p'_j)dF\right| \\
&\leq LSP\left|\int_{\Omega^\infty_{\Sigma_j p_j}}(\Sigma_j p_j - \Sigma_j p'_j)dF + \int_{\Omega^{\Sigma_j p_j}_{\Sigma_j p'_j}}(\Sigma_j p_j - \Sigma_j p'_j)dF\right|
\end{aligned}$$

$$= LSP \left| \int_{\Omega^{\infty}_{\Sigma_j p'_j}} (\Sigma_j p_j - \Sigma_j p'_j) dF \right|$$

$$= LSP \left| \mathbf{P}(\Omega^{\infty}_{\Sigma_j p'_j})(\Sigma_j p_j - \Sigma_j p'_j) \right| \leq LSP |\Sigma_j p_j - \Sigma_j p'_j|$$

$$= LSP |\Sigma_j (p_j - p'_j)| \leq LSP \sqrt{J} \|\mathbf{p}_{ther} - \mathbf{p}'_{ther}\|,$$

where F is the joint distribution of $\hat{M}$ and $\hat{p}_{re}$ and the last inequality follows from the Cauchy-Schwarz inequality, i.e., $(\Sigma_j 1 \cdot x_j)^2 \leq J \Sigma_j x_j^2$. Then, by choosing $L = LSP\sqrt{J}$, the load shedding cost satisfies the condition. Similarly, we can show that the power curtailment cost satisfies the condition by choosing $L = PCP\sqrt{J}$. Then, the expected cost is a Lipschitz continuous function of the action a with $L_a = \sum_{j \in \mathcal{J}} c^{(1)}_{j,fu} + 2c^{(2)}_{j,fu} p^{\max}_j + \sqrt{J}(LSP + PCP)$.

Finally, we conclude that the expected cost is a Lipschitz continuous function for the context $\mathbf{z}$ and the action a, respectively, with $L = \max(L_{\mathbf{z}}, L_a)$.

Appendix B
Proof of Theorem 1

We first introduce some notations and definitions. For each set (hypercube) $p \in \mathcal{P}_{\mathcal{Z}}$, let $\bar{c}_{a,p} := \sup_{\mathbf{z} \in p} c(a, \mathbf{z})$ and $\underline{c}_{a,p} := \inf_{\mathbf{z} \in p} c(a, \mathbf{z})$. For notational brevity, we denote the expected operating cost $c(a(t), \mathbf{z}(t))$ by $c_{a,\mathbf{z}}(t)$. We denote the estimated cost of action a on set p at time period t by $\hat{c}_{a,p}(t)$. Let $\hat{a}(t)$ be the action selected by AP-CLUC at time period t, $a^*(t) = \pi^*_{\mathbf{s}(t)}(\mathbf{z}(t))$ be the best myopic action given unit status $\mathbf{s}(t)$ and context $\mathbf{z}(t)$, and $\bar{a}^*(t)$ be the best myopic action in $\bar{\mathcal{A}}(\mathbf{s}(t), l)$ given unit status $\mathbf{s}(t)$ and context $\mathbf{z}(t)$.

The upper bound on the highest level hypercube that is active at any time t is given by the following lemma.

Lemma 2: In AP-CLUC, all the active hypercubes $p \in \mathcal{P}(t)$ at time t have at most a level of $\lceil \frac{\log_2 t}{\rho} \rceil + 1$.

Proof: Let $l' + 1$ be the level of the highest level active hypercube. We must have $\sum_{l=0}^{l'} 2^{\rho l} < t$, otherwise the highest level active hypercube's level will be less than $l' + 1$. We have for $t > 1$, $\frac{2^{\rho(l'+1)}-1}{2^{\rho}-1} < t \rightarrow 2^{\rho l'} < t \rightarrow l' < \frac{\log_2 t}{\rho}$. ■

With the introduced notations, the one-step regret in time period t is defined as

$$r(t) := c_{\hat{a},\mathbf{z}}(t) - c_{a^*,\mathbf{z}}(t).$$

Consider time period t in which a context $\mathbf{z}(t)$ arrives to level l hypercube denoted by p. Suppose that $m_A(l) = \lceil 2^{l\xi} \rceil$ and the number of previous context arrivals to this hypercube is τ. Note that in AP-CLUC, the estimated costs of all actions in $\bar{\mathcal{A}}(l)$ are virtually updated for every context arrival. Thus, all actions in $\bar{\mathcal{A}}(l)$ are updated τ times. From the one-step regret, we have

$$\begin{aligned} r(t) &= c_{\hat{a},\mathbf{z}}(t) - c_{\bar{a}^*,\mathbf{z}}(t) + c_{\bar{a}^*,\mathbf{z}}(t) - c_{a^*,\mathbf{z}}(t) \\ &\leq c_{\hat{a},\mathbf{z}}(t) - c_{\bar{a}^*,\mathbf{z}}(t) + L\bar{p}_{ther}\sqrt{J}2^{-l\xi}, \end{aligned} \tag{14}$$

where $\bar{p}_{ther} = \max_{j \in \mathcal{J}} [p^{\max}_j - p^{\min}_j]$ and the inequality follows from Lemma 1. Note that there always exists a discretized action $\bar{a}^*$ within the distance $L\bar{p}_{ther}\sqrt{J}^{-l\xi}$ from the optimal action a^*, since the power outputs are uniformly discretized using $m_A(l)$ and the set of feasible power outputs at each time period is a convex set for any given on/off states. Also, we have

$$\hat{c}_{\hat{a},p}(t) \leq \hat{c}_{\bar{a}^*,p}(t) \, a.s.$$

by the action selection rule of AP-CLUC. Then, from (14), we obtain

$$\begin{aligned} r(t) &\leq c_{\hat{a},\mathbf{z}}(t) - \hat{c}_{\hat{a},p}(t) + \hat{c}_{\bar{a}^*,p}(t) - c_{\bar{a}^*,\mathbf{z}}(t) + L\bar{p}_{ther}\sqrt{J}2^{-l\xi} \\ &\leq 2 \max_{a \in \bar{\mathcal{A}}(l)} |c_{a,\mathbf{z}}(t) - \hat{c}_{a,p}(t)| + L\bar{p}_{ther}\sqrt{J}2^{-l\xi}. \end{aligned}$$

Let $\Delta_t := \max_{a \in \bar{\mathcal{A}}(l)} |c_{a,\mathbf{z}}(t) - \hat{c}_{a,p}(t)|$. Then, since the total operating cost is bounded in $[0, 1]$, we have

$$\begin{aligned} \mathbb{E}[r(t)] &\leq 2\mathbb{E}[\Delta_t] + L\bar{p}_{ther}\sqrt{J}2^{-l\xi} \\ &= 2\int_0^1 \mathbf{P}(\Delta_t \geq y) dy + L\bar{p}_{ther}\sqrt{J}2^{-l\xi}. \end{aligned} \tag{15}$$

We also have for all $a \in \bar{\mathcal{A}}(l)$,

$$\mathbb{E}[\hat{c}_{a,p}(t)] - L\sqrt{D}2^{-l} \leq \underline{c}_{a,p} \leq \mathbb{E}[\hat{c}_{a,p}(t)]$$

and

$$\mathbb{E}[\hat{c}_{a,p}(t)] \leq \bar{c}_{a,p} \leq \mathbb{E}[\hat{c}_{a,p}(t)] + L\sqrt{D}2^{-l}$$

from Lemma 1. Thus, we have

$$\begin{aligned} &\{\Delta_t \geq y\} \\ &= \bigcup_{a \in \bar{\mathcal{A}}(l)} \{|c_{a,\mathbf{z}}(t) - \hat{c}_{a,p}(t)| \geq y\} \\ &= \bigcup_{a \in \bar{\mathcal{A}}(l)} \{c_{a,\mathbf{z}}(t) - \hat{c}_{a,p}(t) \leq -y\} \\ &\quad \cup \bigcup_{a \in \bar{\mathcal{A}}(l)} \{c_{a,\mathbf{z}}(t) - \hat{c}_{a,p}(t) \geq y\} \\ &\subset \bigcup_{a \in \bar{\mathcal{A}}(l)} \left\{\underline{c}_{a,\mathbf{z}} - \hat{c}_{a,p}(t) \leq -y\right\} \cup \bigcup_{a \in \bar{\mathcal{A}}(l)} \{\bar{c}_{a,\mathbf{z}} - \hat{c}_{a,p}(t) \geq y\} \\ &\subset \bigcup_{a \in \bar{\mathcal{A}}(l)} \left\{\mathbb{E}[\hat{c}_{a,p}(t)] - L\sqrt{D}2^{-l} - \hat{c}_{a,p}(t) \leq -y\right\} \\ &\quad \cup \bigcup_{a \in \bar{\mathcal{A}}(l)} \left\{\mathbb{E}[\hat{c}_{a,p}(t)] + L\sqrt{D}2^{-l} - \hat{c}_{a,p}(t) \geq y\right\} \\ &\subset \bigcup_{a \in \bar{\mathcal{A}}(l)} \left\{\hat{c}_{a,p}(t) - \mathbb{E}[\hat{c}_{a,p}(t)] \geq y - L\sqrt{D}2^{-l}\right\} \\ &\quad \cup \bigcup_{a \in \bar{\mathcal{A}}(l)} \left\{\hat{c}_{a,p}(t) - \mathbb{E}[\hat{c}_{a,p}(t)] \leq L\sqrt{D}2^{-l} - y\right\}. \end{aligned}$$

By using Hoeffding's inequality, for $y \geq L\sqrt{D}2^{-l}$, we have

$$\mathbf{P}\left(\hat{c}_{a,p}(t) - \mathbb{E}[\hat{c}_{a,p}(t)] \geq y - L\sqrt{D}2^{-l}\right) \leq e^{-2(y - L\sqrt{D}2^{-l})^2 \tau}$$

and

$$\mathbf{P}\left(\hat{c}_{a,p}(t) - \mathbb{E}[\hat{c}_{a,p}(t)] \leq L\sqrt{D}2^{-l} - y\right) \leq e^{-2(y - L\sqrt{D}2^{-l})^2 \tau},$$

since the estimated costs of all actions in $\bar{\mathcal{A}}(l)$ are updated in τ times due to the virtual updates. Note that the effects of the experience reuse on the estimated costs are not considered.

$$\begin{aligned}
&\int_0^1 \mathbf{P}(\Delta_t \geq y)dy \\
&\leq \int_0^{L\sqrt{D}2^{-l}} 1dy + \int_{L\sqrt{D}2^{-l}}^1 \mathbf{P}(\Delta_t \geq y)dy \\
&\leq L\sqrt{D}2^{-l} + 2\cdot 2^{J+l\xi J}\int_{L\sqrt{D}2^{-l}}^1 e^{-2(y-L\sqrt{D}2^{-l})^2\tau}dy \\
&\leq L\sqrt{D}2^{-l} + 2^{(l\xi+1)J+1}\int_{L\sqrt{D}2^{-l}}^1 \frac{dy}{1+2(y-L\sqrt{D}2^{-l})^2\tau} \\
&\leq L\sqrt{D}2^{-l} + 2^{(l\xi+1)J+\frac{1}{2}}\tau^{\frac{-1}{2}}\int_0^{\sqrt{2}\tau^{1/2}(1-L\sqrt{D}2^{-l})} \frac{dx}{1+x^2} \\
&\leq L\sqrt{D}2^{-l} + 2^{(l\xi+1)J+\frac{1}{2}}\tau^{\frac{-1}{2}}\arctan(\sqrt{2}\tau^{1/2}(1-L\sqrt{D}2^{-l})) \\
&\leq L\sqrt{D}2^{-l} + 2^{(l\xi+1)J-\frac{1}{2}}\tau^{\frac{-1}{2}}\pi,
\end{aligned}$$

where the second inequality comes from the fact that $|\bar{\mathcal{A}}(l)| \leq 2^J m_A(l)^J$, the third inequality follows from the fact that $e^{-x} \leq 1/(1+x)$ for $x \geq 0$, and the last inequality follows from the fact that $\arctan(x) \leq \pi/2$ for all $x \in \mathbb{R}$. Plugging the above result to (15), we obtain

$$\mathbb{E}[r(t)] \leq 2L\sqrt{D}2^{-l} + 2^{(l\xi+1)J+\frac{1}{2}}\tau^{\frac{-1}{2}}\pi + L\bar{p}_{ther}\sqrt{J}2^{-l\xi}.$$

If $\tau = 0$, i.e., when hypercube p is selected for the first time, we simply have $\mathbb{E}[r(t)] = 1$. Since p remains active for at most $2^{\rho l}$ arrivals, the total regret in hypercube p, $\mathbb{E}[R_p(T)]$, is bounded by

$$\begin{aligned}
&\mathbb{E}[R_p(T)] \\
&\leq 1 + \sum_{\tau=1}^{2^{\rho l}} 2L\sqrt{D}2^{-l} + 2^{(l\xi+1)J+\frac{1}{2}}\tau^{\frac{-1}{2}}\pi + L\bar{p}_{ther}\sqrt{J}2^{-l\xi} \\
&\leq 1 + 2L\sqrt{D}2^{l(\rho-1)} + 2^{(l\xi+1)J+\frac{\rho l+1}{2}}\pi + L\bar{p}_{ther}\sqrt{J}2^{-l(\xi-\rho)},
\end{aligned}$$

where the last inequality follows from the fact that $\sum_{\tau=1}^{T} \tau^{-1/2} \leq 2T^{1/2}$ [53].

Since the highest time order of each term in the regret is different, in order to minimize the regret bound, we need to optimize the parameters, i.e., ρ and ξ. From the highest time orders of regrets, i.e., $O(2^{l(\rho-1)})$, $O(2^{l\xi J+\frac{\rho l}{2}})$, $O(2^{l(\rho-\xi)})$, we choose the parameters which minimize the regret bound as $\rho = 2(J+1)$, and, $\xi = 1$. Then, with the chosen parameters and Lemma 2, the limiting behaviour of the regret bound in the theorem is given.

APPENDIX C
PROOF OF COROLLARY 1

If the context arrives in the worst-case manner, i.e., uniform context arrival, no level $l+2$ hypercubes become active until all level l hypercubes are deactivated. The number of active hypercubes by time T is maximized in this manner. Then, we have $\sum_{l=1}^{l_{\max}-1} 2^{(D+2(J+1))l} < T$, which implies $l_{\max} < 1 + \log_2 T/(D+2(J+1))$. From Theorem 1, we have

$$\begin{aligned}
R(T) \leq &\sum_{l=1}^{\lfloor 1+\frac{\log_2 T}{D+2(J+1)}\rfloor} \\
&2^{Dl}\left[2^{l(2J+1)}(2L\sqrt{D}+2^{J+\frac{1}{2}}+L\bar{p}_{ther}\sqrt{J})+1\right] \\
&\leq T^{\frac{D+2J+1}{D+2J+2}}2^{D+2J+1}(2L\sqrt{D}+2^{J+\frac{1}{2}}+L\bar{p}_{ther}\sqrt{J}) \\
&\quad + T^{\frac{D}{D+2J+2}}2^{D}.
\end{aligned}$$

REFERENCES

[1] H.-S. Lee, C. Tekin, M. van der Schaar, and J.-W. Lee, "Contextual learning for unit commitment with renewable energy sources," in *Proc. IEEE 5th IEEE Global Conf. Signal Inf. Process.*, 2016, pp. 866–870.

[2] P. A. Ruiz, C. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 642–651, May 2009.

[3] P. Xiong and P. Jirutitijaroen, "A stochastic optimization formulation of unit commitment with reliability constraints," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2200–2208, Dec. 2013.

[4] A. Tuohy, P. Meibom, E. Denny, and M. O'Malley, "Unit commitment for systems with significant wind penetration," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 592–601, May 2009.

[5] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, "A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 431–441, 2011.

[6] A. Papavasiliou, S. S. Oren, and B. Rountree, "Applying high performance computing to transmission-constrained stochastic unit commitment for renewable energy integration," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1109–1120, May 2015.

[7] D. E. Olivares, J. D. Lara, C. A. Cañizares, and M. Kazerani, "Stochastic-predictive energy management system for isolated microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2681–2693, Nov. 2015.

[8] S. Bahramirad and W. Reder, "Islanding applications of energy storage system," in *Proc. IEEE Power Energy Soc. Gen. Meet.*, 2012.

[9] A. Z. Alabedin, "Generation scheduling in microgrids under uncertainties in power generation," *Elect. and Comput. Eng.*, Master's thesis, Univ. Waterloo, Waterloo, ON, Canada, 2012.

[10] H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, "Stochastic energy management of microgrids during unscheduled islanding period," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1079–1087, Jun. 2017.

[11] V. S. Pappala, I. Erlich, K. Rohrig, and J. Dobschinski, "A stochastic model for the optimal operation of a wind-thermal power system," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 940–950, May 2009.

[12] R. Jiang, J. Wang, and Y. Guan, "Robust unit commitment with wind power and pumped storage hydro," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 800–810, May 2012.

[13] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Trans. Power Syst.*, vol. 28, no. 1, pp. 52–63, Feb. 2013.

[14] R. Jiang, J. Wang, M. Zhang, and Y. Guan, "Two-stage minimax regret robust unit commitment," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2271–2282, Aug. 2013.

[15] M. Asensio and J. Contreras, "Stochastic unit commitment in isolated systems with renewable penetration under CVaR assessment," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1356–1367, May 2016.

[16] Y. Xiang, J. Liu, and Y. Liu, "Robust energy management of microgrid with uncertain renewable generation and load," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1034–1043, Mar. 2016.

[17] A. Hussain, V.-H. Bui, and H.-M. Kim, "Robust optimal operation of AC/DC hybrid microgrids under market price uncertainties," *IEEE Access*, vol. 6, pp. 2654–2667, 2018.

[18] B. Zhao, Y. Shi, X. Dong, W. Luan, and J. Bornemann, "Short-term operation scheduling in renewable-powered microgrids: A duality-based approach," *IEEE Trans. Sustain. Energy*, vol. 5, no. 1, pp. 209–217, Jan. 2014.

[19] Y. Wang, Q. Xia, and C. Kang, "Unit commitment with volatile node injections by using interval optimization," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1705–1713, Aug. 2011.
[20] L. Wu, M. Shahidehpour, and Z. Li, "Comparison of scenario-based and interval optimization approaches to stochastic SCUC," *IEEE Trans. Power Syst.*, vol. 27, no. 2, May 2012, Art. no. 913921.
[21] P. B. Luh *et al.*, "Grid integration of intermittent wind generation: A Markovian approach," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 732–741, Mar. 2014.
[22] C. Zhao and Y. Guan, "Unified stochastic and robust unit commitment," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3353–3361, 2013.
[23] Y. Dvorkin, H. Pandzic, M. A. Ortega-Vazquez, and D. S. Kirschen, "A hybrid stochastic/interval approach to transmission-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 621–631, Mar. 2015.
[24] H. Pandžić, Y. Dvorkin, T. Qiu, Y. Wang, and D. S. Kirschen, "Toward cost-efficient and reliable unit commitment under uncertainty," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 970–982, Mar. 2016.
[25] Y. Yu, P. B. Luh, E. Litvinov, T. Zheng, J. Zhao, and F. Zhao, "Grid integration of distributed wind generation: Hybrid Markovian and interval unit commitment," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3061–3072, Nov. 2015.
[26] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE SmartGridComm*, 2011.
[27] L. Yang, M. He, J. Zhang, and V. Vittal, *Spatio-Temporal Data Analytics for Wind Energy Integration*. Berlin, Germany: Springer, 2014.
[28] W. Su, J. Wang, and J. Roh, "Stochastic energy scheduling in microgrids with intermittent renewable energy resources," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1876–1883, Jul. 2014.
[29] 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 J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1219–1234, Jul. 2013.
[30] L. P. Qian, Y. J. A. Zhang, J. Huang, and Y. Wu, "Demand response management via real-time electricity price control in smart grids," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1268–1280, Jul. 2013.
[31] Y. Zhang and M. van der Schaar, "Structure-aware stochastic storage management in smart grids," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 6, pp. 1098–1110, Dec. 2014.
[32] B. G. Kim, Y. Zhang, M. van der Schaar, and J. W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sep. 2016.
[33] C.-L. Hor, S. J. Watson, and S. Majithia, "Analyzing the impact of weather variables on monthly electricity demand," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 2078–2085, Nov. 2005.
[34] G. Singh, "Solar power generation by PV (photovoltaic) technology: A review," *Energy*, vol. 53, pp. 1–13, May 2013.
[35] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 1096–1103, 2007.
[36] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.
[37] C. Tekin and E. Turgay, "Multi-objective contextual bandits with a dominant objective," in *Proc. 27th IEEE Int. Workshop Mach. Learn. Signal Process.*, 2017.
[38] J. Zhu, *Optimization of Power System Operation*. Hoboken, NJ, USA: Wiley, 2009.
[39] N. P. Padhy, "Unit commitment-a bibliographical survey," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1196–1205, May 2004.
[40] L. Faivishevsky and J. Goldberger, "Mutual information based dimensionality reduction with application to non-linear regression," in *Proc. IEEE Mach. Learn. Signal Process.*, Aug. 2010.
[41] A. Slivkins, "Contextual bandits with similarity information," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2533–2568, Jan. 2014.
[42] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3700–3714, Jul. 2015.
[43] C. Tekin and M. van der Schaar, "Active learning in context-driven stream mining with an application to image mining," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3666–3679, 2015.
[44] Pacific & Gas Electric, "Dynamic load profiles in California," [Online]. Available: http://www.pge.com/tariffs/energe _use_prices.shtml
[45] P. Pinson, "Estimation of the uncertainty in wind power forecasting," *Center for Energy. and Processes*, Ph.D. dissertation, Ecole des Mines de Paris, Paris, France, 2006.
[46] A. Y. Saber and G. K. Venayagamoorthy, "Resource scheduling under uncertainty in a smart grid with renewables and plug-in vehicles," *IEEE Syst. J.*, vol. 6, no. 1, pp. 103–109, Mar. 2012.
[47] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, May 2002.
[48] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2002.
[49] L. Song, Y. Xiao, and M. Van Der Schaar, "Demand side management in smart grids using a repeated game framework," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1412–1424, Jul. 2014.
[50] J. Chen and Q. Zhu, "A game-theoretic framework for resilient and distributed generation control of renewable energies in microgrids," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 285–295, Jan. 2017.
[51] A. Mondal, S. Misra, and M. S. Obaidat, "Distributed home energy management system with storage in smart grid using game theory," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1857–1866, Sep. 2017.
[52] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[53] E. Chlebus, "An approximate formula for a partial sum of the divergent p-series," *Appl. Math. Lett.*, vol. 22, no. 5, pp. 732–737, 2009.

Hyun-Suk Lee received the B.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2012 and 2018, respectively. Since 2018, he has been a Postdoctoral Research Associate with the Institute of BioMed-IT, Energy-IT and Smart-IT Technology (BEST), a Brain Korea 21 Plus Program, Yonsei University, Seoul, South Korea. His research interests include communication networks, mobile cloud computing, and smart grid.

Cem Tekin (M'13) received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2008, and the M.S.E. degree in electrical engineering: systems, the M.S. degree in mathematics, and the Ph.D. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2010, 2011, and 2013, respectively. He is currently an Assistant Professor with the Electrical and Electronics Engineering Department, Bilkent University, Ankara, Turkey. From February 2013 to January 2015, he was a Postdoctoral Scholar at University of California, Los Angeles. His research interests include reinforcement learning, multi-armed bandit problems, data mining, multi-agent systems, cognitive radio networks, and smart healthcare. He was the recipient of the University of Michigan Electrical Engineering Departmental Fellowship in 2008 and the Fred W. Ellersick award for the best paper in MILCOM 2009.

Mihaela van der Schaar (F'10) is currently the Chancellor's Professor at UCLA. She has also been the recipient of an NSF Career Award, 3 IBM Faculty Awards, the IBM Exploratory Stream Analytics Innovation Award, the Philips Make a Difference Award and several best paper awards, including the IEEE Darlington Award. She holds 33 granted USA patents.

Jang-Won Lee (M'04–SM'12) received the B.S. degree in electronic engineering from Yonsei University, Seoul, South Korea, in 1994, the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea in 1996, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2004. He is currently a Professor with the School of Electrical and Electronic Engineering, Yonsei University. In 1997–1998, he was with Dacom R&D Center, Daejeon, South Korea. In 2004–2005, he was a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. Since September 2005, he has been with the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea. His research interests include resource allocation, QoS and pricing issues, optimization, and performance analysis in communication networks, and smart grid.