Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 380926, 13 pages
http://dx.doi.org/10.1155/2015/380926

Research Article

Dynamic Energy Storage Control for Reducing Electricity Cost in Data Centers

Shuben Zhang, Jian Yang, Youkang Shi, Xiaomin Wu, and Yongyi Ran

School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

Correspondence should be addressed to Jian Yang; jianyang@ustc.edu.cn

Received 5 June 2014; Accepted 20 October 2014

Academic Editor: Jonathan N. Blakely

As the scale of the data centers increases, electricity cost is becoming the fastest-growing element in their operation costs. In this paper, we investigate the electricity cost reduction opportunities utilizing energy storage facilities in data centers used as uninterrupted power supply units (UPS). Its basic idea is to combine the temporal diversity of electricity price and the energy storage to conceive a strategy for reducing the electricity cost. The electricity cost minimization is formulated in the framework of finite state-action discounted cost Markov decision process (MDP). We apply Q-Learning algorithm to solve the MDP optimization problem and derive a dynamic energy storage control strategy, which does not require any priori information on the Markov process. In order to address the slow-convergence problem of the Q-Learning based algorithm, we introduce a Speedy Q-Learning algorithm. We further discuss the offline optimization problem and obtain the optimal offline solution as the lower bound on the performance of the online and learning theoretic problem. Finally, we evaluate the performance of the proposed scheme by using real workload traces and electricity price data sets. The experimental results show the effectiveness of the proposed scheme.

1. Introduction

Cloud computing is an emerging Internet-based computing paradigm which offers on-demand computing services to cloud consumers. To meet the increasing demands of computing and storage resources in cloud computing, there is an increasing trend toward large-scale data centers. As more data centers are deployed and their scale increases, energy consumption cost is becoming the fastest-growing element in their operation costs, including the computing energy cost, cooling energy cost, and other energy overheads. It has been estimated that energy consumption cost may amount to 30%–50% percentage of operation cost of large-scale data centers built by companies such as Google, Microsoft, and Facebook [1]. In fact, data centers consumed approximately 1.5% of all electricity consumption worldwide in 2010, which was about 56% higher than the preceding five years [2, 3]. In the near future, the energy consumption cost problem of data centers is likely to worsen and be more challenging since the technology infrastructures emerge and upgrade from a recessionary period. Hence, efficiently controlling the electricity cost of data centers has attracted an intensive concern of broader research community participating from both academia and industry in the recent years.

As we know, electricity cost generation depends not only on the total amount of energy consumed by the data centers, but also on the electricity price. Therefore, the electricity price is also an important factor in the electricity cost of data centers. With the development of smart grid technology which is a technology for the next generation power grid, more and more electricity markets are undergoing deregulation where the electricity market operators offer dynamic electricity rates to large industrial and commercial customers instead of traditional flat rates at the retail level. Thus, there is an opportunity for us to achieve the electricity consumption cost saving in data centers by observing and utilizing the time-varying electricity price in the deregulated electricity markets.

Normally, the UPS units may be deployed in data centers, and provide emergency energy to power them up using stored energy before the backup diesel generators (DG) can start up and operate as a secondary power source when the main

power system experiences an outage. Usually, the transition from the main power system to the secondary power source takes 10–20 seconds. As an improvement of the rechargeable battery, the UPS units have enough energy storage capacity for keeping a data center working 5–30 minutes at its maximum power demand [4]. Hence, the excessive energy storage capacity gives us a good opportunity for electricity cost saving utilizing the UPS units to dynamically control energy storage.

Based on the above two facts, the basic principle for achieving the electricity cost saving is recharging the UPS units residing in the data center when the outside electricity price is low and discharging for powering the data center when the outside electricity price is high. Hence, this paper focuses on a dynamic energy storage control strategy for reducing the electricity cost of the data centers. Dynamic energy storage control is expected to adapt the fluctuation of the electricity price and the workload by dynamically making recharge/discharge decisions for the UPS units. It aims for achieving substantial electricity cost saving without performance degradation.

In this paper, we formulate the electricity cost reduction problem utilizing energy storage facilities as the discounted cost Markov decision process. Since the statistical information about the workload arrival and the electricity price is not available, we propose an online algorithm based on Q-Learning and Speedy Q-Learning approaches to solve the optimization problem. Particularly, the main contributions of this paper are summarized as follows.

(i) The problem of electricity cost minimization in data centers with energy storage facilities for time-varying electricity prices under deregulated electricity markets is modeled by a discounted cost Markov decision process, which achieves the cost saving by making decisions to recharge/discharge the battery.

(ii) In order to solve the optimization problem, we propose a dynamic energy storage control strategy based on the Q-Learning algorithm, which avoids the reliance on any prior knowledge of the workload and the electricity prices. Furthermore, we introduce a Speedy Q-Learning algorithm to accelerate convergence of the standard Q-Learning.

(iii) We formulate an offline optimization problem of electricity cost minimization for obtaining the optimal offline solution as the lower bound on the performance of the online and learning theoretic problem. The offline optimization problem is solved by mapping it into a tractable mixed integer linear programming instead of nonlinear programming.

(iv) Finally, the experiments are carried out based on real workload traces and electricity price data sets to show the performance of the proposed scheme. By using the real traces that may not provably follow the Markovian assumption, the result also shows that the proposed scheme generally performs well.

The rest of the paper is organized as follows: in Section 2 some related works in this area are presented and discussed. Section 3 describes a system model for energy management system using energy storage facilities in date centers. Section 4 formulates the problem of electricity cost consumption in the data centers with energy storage facilities as a discounted cost Markov decision process. Section 5 is devoted to designing a dynamic energy storage control strategy of battery based on Q-Learning and Speedy Q-Learning algorithms to solve the optimization problem. The optimal offline solution is discussed in Section 6. In Section 7, we provide the numerical evaluation results and performance comparisons. Finally, conclusions are drawn in Section 8.

2. Related Work

The severe energy consumption problem in data centers has motivated many works on reducing their electricity cost. These works may be roughly categorized into two basic types of mechanisms: (1) reduce the energy consumption or improve the energy efficiency of the data centers; and (2) exploit the temporal and geographical variation of electricity prices to achieve the electricity cost saving.

Regarding the first mechanism, new hardware designs and engineering techniques such as energy-efficient chips, multicore servers [5], DC power supplies [6], advanced cooling systems [7, 8], and virtualization [9, 10] have been developed in order to improve the power utilization efficiency (PUE) of data centers. From the perspective of algorithm design, the energy consumption saving can operate at two different levels: the server level and the data center level [4]. At the server level, dynamic voltage-frequency scaling (DVFS) [11] offers a way to reduce power consumption by adapting both voltage and frequency of CPU with respect to changing workloads. However, DVFS can be applicable only for components (like CPU) that support multiple speed and voltage levels. DVFS based power saving policies can be found in [12, 13]. Dynamic power management (DPM) is another energy conservation approach, which turns off the power or switches the system to a low-power state when inactive. It can be employed for any system component with multiple power states. In [14], DPM is applied to achieve energy-efficient computation by selectively turning off (or reducing the performance of) system components when they are idle (or partially unexploited).

At the data center level, dynamic cluster reconfiguration (DCR) [15], VM migration and consolidation for load balancing and power management [16], and so forth, approaches are widely discussed for reducing energy consumption in the data centers. DCR in [15] develops an online measurement based algorithm to decide the number of servers to power on/off to achieve energy saving while keeping the overload probability below a desired threshold, which makes a decision without any prior knowledge of the workload statistics. VM migration and consolidation [16] achieve energy saving by continuous consolidation of VMs according to current resource utilization, virtual network topologies connecting VMs, and thermal state of computing nodes. These meth-

ods mentioned above mainly focus toward reducing energy consumption to save electricity cost. They can operate as a complementary way to assist the method proposed in this paper to further reduce the electricity cost.

The second mechanism for reducing electricity cost relies on the fact of the notable temporal and geographical variations in electricity prices. In [1], Qureshi et al. develop and analyze a new method for reducing the electricity costs when running large Internet-scale systems. The key idea of the method is to distribute more traffic to data centers with low electricity price. In [17], Rao et al. utilize both the location diversity and the time diversity of electricity prices in the multiple electricity markets environment to minimize the total electricity cost while guaranteeing the quality of service (QoS). Luo et al. [18] propose a novel spatiotemporal load balancing approach to leverage both geographic and temporal variations of electricity price to minimize energy cost for distributed internet data centers (IDC). However, those works mentioned above do not utilize energy storage facilities residing in data centers, which may be used to achieve further electricity cost saving. Compared with existing techniques for electricity cost reduction, the methods of energy storage have no performance degradation of the data center. In this paper, our work focuses on the problem of electricity cost minimization in data centers with energy storage facilities under deregulated electricity markets where the electricity prices exhibit temporal variation, which is mainly motivated by [19]. In [19], an online control algorithm using Lyapunov optimization theory is proposed for reducing the time average electric utility bill in a data center, and the solution has the threshold structure. Although simple, the technique of Lyapunov optimization is unable to learn the system dynamics, which may not lead to an optimal control of energy storage. Alternatively, by exploiting a Markov decision process approach and reinforcement learning tool, the proposed algorithms learn the system dynamics and adapt the control decision accordingly for saving more electricity cost. Generally, the optimal control policies for Markov decision process suffer from the "curse of dimensionality." In our work, we consider the total energy consumption of all components in the data center as the energy consumption state instead of each component's individually. Furthermore, there are only three actions on the battery, that is, recharging, discharging, and doing neither. Thus, all of those considerations may effectively alleviate the problem of "curse of dimensionality."

3. System Model

In this section, we describe system architecture model for energy management in data center, present the models for battery, energy consumption, and electricity cost, as well as formulating the problem of dynamic energy storage control to minimize the expected total electricity cost.

3.1. System Architecture. A general system architecture model for data center with energy storage facilities, depicted in Figure 1, is composed of an energy management system (EMS) and a data center facility. EMS acts as the heart of the energy management framework and manages the energy provision in data center, while the data center facility provides computation and storage resources for executing the submitted tasks. In EMS, the key components include information collector (IC) and energy storage management unit (ESMU). IC is to collect the information of the electricity prices, energy storage, and the energy demand generated by the data center periodically, while ESMU is to make the optimal decision on whether recharging or discharging the energy storage facilities for electricity cost minimization according to the information collected by IC. The energy storage unit (ESU), that is, UPS, has the capability of storing energy drawn from the power grid and discharging the stored energy to power the data center. Below, we use the terms UPS and battery interchangeably. The main work of this paper is to propose a dynamic energy storage control strategy for ESMU.

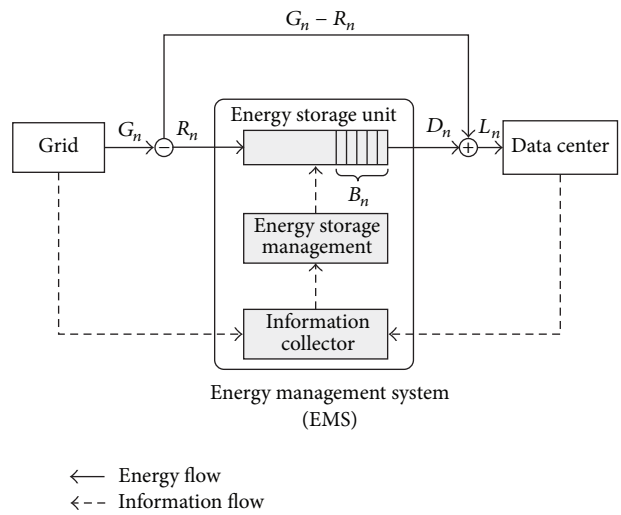

Figure 1: Energy management framework using energy storage facilities in data center.

The basic running process of EMS can be generally described as follows. IC periodically collects the battery level information as well as the electricity price information from the grid. The data center submits its energy demand information to IC, and ESMU uses this information to make the decision that the energy supply draws from grid or the battery. Finally, the data center can provide services using the energy managed by EMS.

3.2. Mathematical Model. In this subsection, we introduce the time-slotted system model used in this paper, and the time is divided into slots of equal duration of m minutes. It should be noted that small value of the time slot size, m, is beneficial for characterizing the state variation of the system in a small time granularity, thus achieving a better cost saving policy due to its prompt adaptation to the changes of the system state. But it may increase the battery cost owing to the increased switching frequency switching of recharge/discharge battery. Therefore, a time slot size should be appropriately selected. The energy storage control

decisions are made at the beginning of each slot, and the system's state is assumed to be constant throughout each slot.

From [4], we know that the energy consumption demand of data center in each slot is proportional to the total number of workload requests needed to be served in that slot. The workload requests served in each slot consist of the unfinished requests in the last slot and the new incoming requests in current slot, which implies that the energy consumption demand of data center in each slot depends upon the previous energy demand, not upon other history demands, and it fulfills the Markov property. Thus, we model the energy consumption demands of data center in each slot as correlated time processes following a first-order discrete-time Markov model. The energy consumption demand in each slot is assumed to be known at the beginning of a time slot. In reality, this has to be estimated. There are several effective methods for estimating the workload, such as autoregressive and moving-average (ARMA). Let L_n be energy consumption demand of data center in the slot n, $L_n \in \mathscr{L} \triangleq \{l_1, l_2, \ldots, l_{N_{\mathscr{L}}}\}$, where $N_{\mathscr{L}}$ is the number of elements in $\mathscr{L}$. The elements in $\mathscr{L}$ have nonnegative and finite values; that is, $0 \le l_i \le l_{\max}$, for $i = 1, \ldots, N_{\mathscr{L}}$. $r_l\ (l_i, l_j)$ denotes the state transition probability which means that the probability of state transition from l_i to l_j is $r_l\ (l_i, l_j)$.

The energy market usually consists of Day-Ahead market and Real-Time market [1]. In this paper, we consider the data centers in Real-Time market. Real-time market is a spot market in which the current real-time price is calculated every five minutes or so, based on actual grid operating conditions, rather than expected load. The electricity price in Real-Time electricity market in the slot n is denoted by P_n, where $P_n \in \mathscr{P} \triangleq \{p_1, p_2, \ldots, p_{N_{\mathscr{P}}}\}$, $N_{\mathscr{P}}$ is the number of elements in $\mathscr{P}$, and the elements are assumed to be nonnegative and finite values; that is, $0 \le p_i \le p_{\max}$, for $i = 1, \ldots, N_{\mathscr{P}}$. Following [20], we model the electricity price P_n as a Markov chain, and $r_p\ (p_i, p_j)$ denotes its state transition probability.

In current data centers, UPS units use lead-acid batteries typically. There are several characteristics of battery operation when using a lead-acid battery practically. For a given battery, each recharge-discharge cycle has energy loss due to AC-DC conversion, so the battery may not be completely efficient, and its performance is affected by the recharge efficiency $\eta_c \in (0, 1]$ and discharge efficiency $\eta_d \in (0, 1]$ [21]. The energy in the battery is also subject to dissipation over time; it exhibits a leaky character. However, considering that storage leak loss is much smaller than that of interest to us, it is negligible for lead-acid batteries [19]. The recharging rate R is assumed to be constant. This is a reasonable assumption when the battery recharges in the constant current way. To assess the impact of repeated recharging and discharging on the battery's lifetime, we assume that each recharge and discharge operation incurs a fixed cost of C_r and C_d, respectively. From [19], we have $C_r = C_d = C_b/K$ when a new battery costs C_b dollars and it can sustain K recharge/discharge cycles. Let B_n be the battery energy level in the slot n, which is no more than battery capacity of $B_{\max}$; that is, $B_n \le B_{\max}$ for all n. The UPS unit is mainly employed to power data center using the stored energy in case of power failure before the backup diesel generators start up and provide power. In order to ensure the reliability of the data center, the battery energy level B_n is required to maintain a minimum energy level $B_{\min} \ge 0$; that is, $B_n \ge B_{\min}$ for all n. Hence, the battery energy level B_n is subject to a constraint:

$$B_{\min} \le B_n \le B_{\max}. \tag{1}$$

Let $X_n \in \{-1, 0, 1\}$ be the decision variable of the event that the battery is recharged/discharged in the slot n. Without loss of generality, we assume that recharge/discharge operations cannot be done simultaneously; that is to say, we can either recharge or discharge the battery or do neither, but not both. Thus, X_n can be defined as follows:

$$X_n = \begin{cases} 1, & \text{if recharging the battery in the slot } n \\ -1, & \text{if discharging the battery in the slot } n \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let R_n represent the amount of energy bought to recharge the battery in the slot n, and D_n denote the energy used towards satisfying demand in the slot n. Then, R_n and D_n can be expressed as follows:

$$\begin{aligned} R_n &= \delta(X_n) R, \\ D_n &= \delta(-X_n) L_n, \end{aligned} \tag{3}$$

where $\delta(x)$ is an indicator function, defined as

$$\delta(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The update equation for the battery energy level B_{n+1} in the slot $n+1$ can be expressed as

$$\begin{aligned} B_{n+1} &= B_n + \eta_c R_n - \frac{1}{\eta_d} D_n \\ &= B_n + \eta_c \delta(X_n) R - \delta(-X_n) \frac{1}{\eta_d} L_n, \end{aligned} \tag{5}$$

where $\eta_c R_n$ implies that the energy purchased to recharge the battery is reduced by the recharge efficiency η_c, while $(1/\eta_d) D_n$ implies that only a fraction η_d of the discharged energy is converted into electricity under the discharge efficiency η_d.

According to inequality (1), the battery level B_{n+1} cannot exceed its maximum capacity and be lower than the minimum battery level. Therefore, R_n and D_n have to satisfy the constraints as follows:

$$\begin{aligned} 0 &\le \eta_c R_n \le B_{\max} - B_n, \\ 0 &\le \frac{1}{\eta_d} D_n \le B_n - B_{\min}. \end{aligned} \tag{6}$$

Let G_n represent the external energy drawn from the power grid in the slot n, which is used to power data center and recharge the battery. As shown in Figure 1, in order to

meet the energy consumption demand for powering the data center in the slot n, we have

$$L_n = G_n - R_n + D_n. \tag{7}$$

Thus, the total amount of energy drawn from the grid in the slot n can be written as

$$\begin{aligned} G_n &= L_n - D_n + R_n \\ &= L_n - \delta(-X_n) L_n + \delta(X_n) R. \end{aligned} \tag{8}$$

For notational simplicity, according to the indicator function $\delta(x)$, G_n can also be denoted as

$$G_n = \delta(1 + X_n) L_n + \delta(X_n) R. \tag{9}$$

Define C_n as the total immediate cost incurred in the slot n. Then, we have for all n

$$\begin{aligned} C_n &= P_n G_n + \delta(X_n) C_r + \delta(-X_n) C_d \\ &= P_n (\delta(1 + X_n) L_n + \delta(X_n) R) \\ &\quad + \delta(X_n) C_r + \delta(-X_n) C_d, \end{aligned} \tag{10}$$

where the term $P_n G_n$ in the first equation is the electricity cost for the energy consumption in the slot n, while the term $\delta(X_n)C_r + \delta(-X_n)C_d$ represents the battery cost for each recharge and discharge operation.

In this paper, the goal of dynamic energy storage control is to minimize the expected total electricity cost in the data centers with energy storage facilities. Based on the above models, the problem can be formulated as follows:

$$\begin{aligned} &\min_{\{X_i\}_{i=0}^{\infty}} \quad \lim_{N \to \infty} \mathbb{E}\left\{\sum_{n=0}^{N} \gamma^n C_n\right\} \\ &\text{s.t.} \qquad (5) \text{ and } (6), \end{aligned} \tag{11}$$

where $\mathbb{E}[\cdot]$ denotes expectation operator, and $0 < \gamma < 1$ is the discount factor that represents value reduction over time. The reason for considering discounted electricity costs is to emphasize early decisions and costs, in order to emulate the effect of reduced battery efficiency over time. Note that the total discounted electricity cost is finite, since the per-slot costs are bounded. We call this problem the *expected total electricity cost minimization problem* (ETC-problem) as the data center aims at minimizing the total electricity cost.

According to (10), (11) can be rewritten as

$$\begin{aligned} &\min_{\{X_i\}_{i=0}^{\infty}} \quad \lim_{N \to \infty} \mathbb{E}\left\{\sum_{n=0}^{N} \gamma^n \left[P_n (\delta(1 + X_n) L_n + \delta(X_n) R)\right.\right. \\ &\qquad \left.\left. + \delta(X_n) C_r + \delta(-X_n) C_d\right]\right\} \\ &\text{s.t.} \qquad (5) \text{ and } (6). \end{aligned} \tag{12}$$

3.3. Discussion. In data center, the lower-level management routines like server consolidation and instantiation of new VMs may be executed. Different management routines may have different demand profiles of energy consumption. But once the lower-level management routine is given, the demand profile for the workload is determined and can be mathematically modeled. Hence, we can still apply the above mentioned model to achieve the electricity cost saving.

4. Cost Management Problem as an MDP

In this section, we will map the problem (12) into the framework of Markov decision process (MDP). A Markov decision process, also referred to as a discrete time stochastic control process, provides a mathematical framework for modeling decision-making situations where outcomes are partly random and partly under the control of the decision maker [22]. An MDP can be defined via a 4-tuple $\langle \mathcal{S}, \mathcal{A}, p_x(s, s'), C_x(s, s') \rangle$, where

(i) $\mathcal{S}$ is the finite set of states,

(ii) $\mathcal{A}$ is the finite set of actions,

(iii) $r_x(s, s') = \Pr(s_{n+1} = s' \mid s_n = s, x_n = x)$ denotes the probability that the system is in state $s' \in \mathcal{S}$ at the $(n+1)$th slot when the decision maker chooses action $x \in \mathcal{A}$ in state s at the nth slot,

(iv) $C_x(s, s')$ denotes the immediate cost yielded when the state of the system at the nth slot is s, action $x \in \mathcal{A}$ is selected, and the system occupies state s' at the $(n + 1)$th slot.

The energy management system in data center, as described above, can be formulated as a finite-state discrete-time MDP. In the model, let S_n denote the joint state (hereafter *state*) of the system at the nth slot, and S_n consists of the energy consumption demand L_n, the battery energy level B_n, and the electricity price P_n. Thus S_n can be expressed as the triple (L_n, B_n, P_n). Since all components of S_n are discrete and finite, the number of elements in $\mathcal{S}$ is finite, and the set of states can be denoted by $\mathcal{S} = \{s_1, s_2, \ldots, s_{N_\mathcal{S}}\}$, where $N_\mathcal{S}$ is the number of elements in $\mathcal{S}$. An action set represents all allowable actions in all possible states. According to the definition of X_n in (2), let $\mathcal{A} = \{-1, 0, 1\}$ be the set of actions for the system, where action -1 indicates that the battery is discharged while action 1 indicates that the battery is recharged, and action 0 indicates that the battery is neither recharged nor discharged. A policy specifies the decision rule to be used at all decision epoches, and here the time of making decision-making is referred to as decision epoches. The policy provides the decision maker with a prescription for action selection under any possible future system state or history. The policy $\pi = \{\pi_i, i \geq 0\}$ maps the state space to the action space. In this paper, we restrict our attention to stationary deterministic policies that do not depend on time but only on the current state. Let $r_{x_k}(s_i, s_j)$ denote the transition probability from state s_i to state s_j when action x_k is taken. The immediate electricity cost is $C_{x_k}(S_n, S_{n+1})$ when the action $x_k \in \mathcal{A}$ is taken in state S_n at the nth slot, and

then the state changes to S_{n+1} at the $(n+1)$th slot. Thus, the objective of an MDP is to find the optimal energy storage control policy $\pi(\cdot) : \mathcal{S} \rightarrow \mathcal{A}$ that minimizes the expected total discounted electricity cost for the energy consumption in the data center over an infinite time horizon. Here the immediate cost function can be expressed as

$$C_x(S_n, S_{n+1}) = P_n(\delta(1+X_n)L_n + \delta(X_n)R) + \delta(X_n)C_r + \delta(-X_n)C_d \tag{13}$$

and the expected total discounted electricity cost is equivalent to (12), and $X_n = \pi(S_n)$ is the action taken when the system is in state S_n.

As described in Section 3, the energy consumption demand and the electricity price can be described by the state transition probability functions, while the battery energy level B_{n+1} can be uniquely derived by the update equation (5) under the given policy π and the current system state S_n. Since the system state consists of the energy consumption, the battery energy level, and electricity price, the transition of the system state depends only on the current state and the current action. This means that the model described above fulfills the Markov property which indicates that a state depends only on the previous state not on more previous states. Thus, we can make use of dynamic programming (DP) and reinforcement learning (RL) theories to solve the problem (12). For convenience, we will introduce the definition of the *state-value function* and *action-value function* before solving the MDP problem [23].

Being in search of an optimal policy, the decision maker needs a facility to differentiate the desirability of possible successor states, in order to decide on the best action. A common way to rank states is by computing and using a so-called state-value function which estimates the expected discounted sum cost when starting in a specific state s_i and taking actions determined by policy π. Accordingly, the state-value function for policy π is defined as follows:

$$V^{\pi}(s_i) \triangleq \sum_{s_j \in \mathcal{S}} r_{\pi(s_i)}(s_i, s_j)\left[C_{\pi(s_i)}(s_i, s_j) + \gamma V^{\pi}(s_j)\right]. \tag{14}$$

Equation (14) is also called *the Bellman equation for* V^{π}, and it expresses a relationship between the value of a state and the value of its successor states.

Similarly, define the value of taking action x_k in state s_i under the policy π, denoted by $Q^{\pi}(s_i, x_k)$, as the expected discounted cost starting from s_i, taking the action $x_k \in \mathcal{A}$, and thereafter following policy π. $Q^{\pi}(s_i, x_k)$ is expressed as

$$Q^{\pi}(s_i, x_k) \triangleq \sum_{s_j \in \mathcal{S}} r_{x_k}(s_i, s_j)\left[C_{x_k}(s_i, s_j) + \gamma V^{\pi}(s_j)\right]. \tag{15}$$

Q^{π} is referred to as action-value function for policy π.

For finite MDPs, an optimal policy can be precisely defined in the following way. A policy π is defined to be better than or equal to a policy π' if its expected discounted cost of π is less than or equal to that of π' for all states. In other words, $\pi \geq \pi'$ if and only if $V^{\pi}(s_i) \leq V^{\pi'}(s_i)$ for all $s_i \in \mathcal{S}$. Let π^* be the optimal policy which is better than or equal to all the other policies. Accordingly, the state-value function under the optimal policy π^* is

$$V^{\pi^*}(s_i) = \min_{x_i \in \mathcal{A}} Q^{\pi^*}(s_i, x_i). \tag{16}$$

Intuitively, (16) expresses the fact that the value of a state under the optimal policy π^* must equal the expected discounted cost for the best action from that state. So we can see that the optimal policy is the *greedy policy*. According to (16), the optimal action-value function $Q^{\pi^*}(s_i, x_i)$ under the optimal policy π^* can be written as

$$Q^{\pi^*}(s_i, x_k) = \sum_{s_j \in \mathcal{S}} r_{x_k}(s_i, s_j)\left[C_{x_k}(s_i, s_j) + \gamma \min_{x_i \in \mathcal{A}} Q^{\pi^*}(s_j, x_i)\right]. \tag{17}$$

As seen from the above analysis, in order to minimize the expected total electricity cost, we can obtain the optimal policy by learning Q-value, instead of estimating the demand and real-time electricity prices to solve (11) directly. Thus, solving the ETC-problem requires the prior information on the values of $r_{x_k}(s_i, s_j)$. Unfortunately, accurate probability distribution of state transition is usually difficult to be known beforehand in practice. Consequently, V^{π^*} and π^* cannot be computed using value iteration. To overcome this difficulty, we consider applying a *model-free* learning theoretic algorithm based on RL to arrive at an optimal policy π^* which minimizes the expected discounted total cost by taking actions and observing their corresponding costs. In the next section, the detailed learning theoretic algorithm is presented.

5. Learning Theoretic Algorithm

In this section, we will introduce learning theoretic algorithms, namely, Q-Learning and Speedy Q-Learning, which we have used to find optimal energy storage control policy. Q-Learning is a reinforcement learning (RL) algorithm for solving the MDP problems and it directly estimates Q^{π^*} under the assumption that the system's dynamics are completely unknown a priori. It is a well-known model-free algorithm, so the main advantages of the algorithm are simple and easy implementation as well as online operation [24]. Hence, Q-Learning is well-suited for our ETC-problem. The core of Q-Learning algorithm is a Q-table and an algorithm for updating the Q-table and choosing actions. A Q-table $Q(s_i, x_k)$ is a matrix indexed by state s_i and action x_k, which is the expected discounted cost of taking action x_k in state s_i.

According to (15), we can see that the action-value function $Q^{\pi}(s_i, x_k)$ can be expressed as a combination of the expected immediate cost and the state-value function $V^{\pi}(s_j)$ of the next state when following the policy π. Note that $Q^{\pi}(s_i, x_k)$ provides the expected long-term consequences for each state-action pair. Then the action incurring the lowest expected cost can be taken as the optimal action just by observing $Q^{\pi^*}(s_i, x_k)$. Hence, the optimal action-value function allows optimal actions to be selected without

knowing anything about $r_{x_k}(s_i, s_j)$, and we can derive the optimal policy π^* by estimating $Q^{\pi^*}(s_i, x_k)$. The Q-Learning process tries to find $Q^{\pi^*}(s_i, x_k)$ in a recursive manner. Let $Q_n(s_i, x_k)$ be the estimate of $Q^{\pi^*}(s_i, x_k)$ in the nth iteration. Then, in each slot the update process of the estimate $Q_n(s_i, x_k)$ can be described as follows:

(i) observe the current state $s_i \leftarrow S_n \in \mathcal{S}$,

(ii) choose action $x_k \leftarrow X_n \in \mathcal{A}$, and then perform the chosen action x_k,

(iii) observe the next state $s_j \leftarrow S_{n+1} \in \mathcal{S}$, and receive an immediate cost $C_{x_k}(s_i, s_j)$,

(iv) update the estimate $Q_n(s_i, x_k)$ according to

$$\begin{aligned} Q_n(s_i, x_k) = {} & (1-\rho_n) Q_{n-1}(s_i, x_k) \\ & + \rho_n \left[C_{x_k}(s_i, s_j) + \gamma \min_{x_i \in \mathcal{A}} Q_{n-1}(s_j, x_i) \right], \end{aligned} \tag{18}$$

where ρ_n is the learning rate in the nth iteration, and it is responsible for weighing the newly learnt experience. The sequence $Q_n(s_i, x_k)$ can be proven to converge with probability 1 to $Q^{\pi^*}(s_i, x_k)$ as $n \to \infty$ when ρ_n satisfies the *stochastic approximation conditions* $0 < \rho_n < 1$ and further, $\sum_n \rho_n = \infty$, $\sum_n \rho_n^2 < \infty$ [25]. $Q_0(s_i, x_k)$ can be initialized arbitrarily for all $(s_i, x_k) \in \mathcal{S} \times \mathcal{A}$.

Based on the above discussion, the estimate $Q_n(s_i, x_k)$ can be used for determining an action. However, the optimal action is determined depending on the accurate estimate for $Q^{\pi^*}(s_i, x_k)$. Otherwise, there will always be cases that the actions with current minimum cost are not producing the real lowest cost return. During the learning process, unguided randomized exploration cannot guarantee acceptable performance, while taking greedy actions exploiting the available information in $Q_n(s_i, x_k)$ can guarantee a certain level of performance, but exploiting what is already known about the system prevents the discovery of better actions. In order to estimate $Q^{\pi^*}(s_i, x_k)$ accurately, the action selection method should harmonize the trade-off between exploitation and exploration such that EMS can reinforce the evaluation of the actions it already knows to be good but also explore new actions. Here, we consider the ϵ-greedy method. This method selects a random action (explores) with probability ϵ and the best action (exploits), that is, the one that has the lowest Q-value at the moment, with probability $1-\epsilon$ at each slot, where $0 < \epsilon < 1$. Therefore, exploration probability ϵ provides Q-Learning to be able to continuously explore itself in the new environment for other possibilities of actions despite of the current lowest cost.

Although it has been shown that the sequence $Q_n(s_i, x_k)$ converges to the optimal action-value function $Q^{\pi^*}(s_i, x_k)$, Q-Learnging suffers from slow-convergence when the discount factor γ is close to one. To address this problem, asynchronous Speedy Q-Learning (ASQL) method is applied to improve the convergence rate. At each slot step, ASQL uses two successive estimates of the action-value function to update the Q-values for achieving a faster convergence rate than standard Q-Learning. The update process for ASQL is described as follows:

$$\begin{aligned} Q_{n+1}(s_i, x_k) = {} & Q_n(s_i, x_k) \\ & + \rho_n \left(\widehat{Q}_{\text{est1}}(s_i, x_k) - Q_n(s_i, x_k) \right) \\ & + (1-\rho_n) \left(\widehat{Q}_{\text{est2}}(s_i, x_k) - \widehat{Q}_{\text{est1}}(s_i, x_k) \right), \end{aligned} \tag{19}$$

where the action x_k is chosen in state s_i using the ϵ-greedy exploration method, and the system occupies state s_j next. Let $\widehat{Q}_0(s_0, x_0) = \widehat{Q}_{-1}(s_0, x_0) = 0, \forall s_0 \in \mathcal{S}$ and $x_0 \in \mathcal{A}$. Then, $\forall k \geq 0$, $\widehat{Q}_{\text{est1}}(s_i, x_k)$ and $\widehat{Q}_{\text{est2}}(s_i, x_k)$ are calculated, respectively, by

$$\widehat{Q}_{\text{est1}}(s_i, x_k) = C_{x_k}(s_i, s_j) + \gamma \min_{x_i \in \mathcal{A}} Q_{n-1}(s_j, x_i), \tag{20}$$

$$\widehat{Q}_{\text{est2}}(s_i, x_k) = C_{x_k}(s_i, s_j) + \gamma \min_{x_i \in \mathcal{A}} Q_n(s_j, x_i). \tag{21}$$

In the ASQL algorithm, let ρ_n decay linearly with time; that is, $\rho_n = 1/(n+1)$, where n is the number of learning iteration. Note that other (polynomial) learning steps can also be used with Speedy Q-Learning. However, it has been shown that the rate of convergence of ASQL is optimized for $\rho_n = 1/(n+1)$ [26]. Intuitively, the third term in the right-hand side of (19) does not play a role for small n, $\rho_n \approx 1$, and the aggressive steps are taken as n increases when the error in the estimate $\widehat{Q}_{\text{est2}}(s_i, x_k) - \widehat{Q}_{\text{est1}}(s_i, x_k)$ is large. Further, when n is very large, the error of the estimate goes to zero as Q_n approaches its optimal value Q^*, and then there has $\widehat{Q}_{\text{est2}}(s_i, x_k) \approx \widehat{Q}_{\text{est1}}(s_i, x_k)$, thus the third term does not affect the updates.

By applying the proposed scheme, we can obtain the optimal energy storage control policy using storage facilities in data centers for electricity cost minimization. The more detailed procedures of the proposed scheme are presented in Algorithm 1.

6. Optimal Offline Solution

In this section, we give a lower bound on the performance of the learning theoretic problem by the optimal offline solution, which is employed as a benchmark to evaluate the optimality of the proposed learning theoretic algorithm. In order to formulate the offline optimization problem, we assume that all the future workload arrivals as well as the electricity price variations are known noncausally before the decisions of energy storage control are made. This information can be obtained from the traces of the workload and electricity price in advance. Online learning theoretic problem optimizes the expected total electricity cost over an infinite horizon while the offline solution does that over a realization of the MDP for a finite number of time slots. As previously described, an MDP realization is a sequence of state transitions of the workload, the battery energy level and the electricity price state processes for a finite number of time slots. Hence, we can optimize X_n such that the expected total electricity cost is

(1) **Initialize:**
for each $s_i \in \mathcal{S}$, $x_k \in \mathcal{A}$ **do**
  Initialize $Q(s_i, x_k)$ arbitrarily
end for
Initialize learning counter $n \leftarrow 0$
Initialize starting state $s_i \leftarrow S_n \in \mathcal{S}$
(2) **Learning:**
repeat
  Decide to explore/exploit action with probability ϵ
  if exploration **then**
    Choose action $x_k \in \mathcal{A}$ at random
  else if exploitation **then**
    choose action $x_k = \arg_{x_i} \log_{x_i \in \mathcal{A}} Q(s_i, x_i)$
  end if
  Take action x_k
  Observe the next state $s_j \leftarrow S_{n+1} \in \mathcal{S}$
  Receive an immediate cost $C_{x_k}(s_i, s_j)$
  Calculate $\widehat{Q}_{\text{est1}}(s_i, x_k)$ according to (20)
  Calculate $\widehat{Q}_{\text{est2}}(s_i, x_k)$ according to (21)
  Update the $Q(s_i, x_k)$ estimate as follows:
  $$\begin{aligned} Q_{n+1}(s_i, x_k) \leftarrow{} & Q_n(s_i, x_k) \\ & + \rho_n \left(\widehat{Q}_{\text{est1}}(s_i, x_k) - Q_n(s_i, x_k)\right) \\ & + (1 - \rho_n)\left(\widehat{Q}_{\text{est2}}(s_i, x_k) - \widehat{Q}_{\text{est1}}(s_i, x_k)\right) \end{aligned}$$
  Update the current state $s_i \leftarrow s_j$
  Update learning counter $n \leftarrow n + 1$
until $n = N_L$

Algorithm 1: Dynamic energy storage control strategy based on Q-Learning and Speedy Q-Learning algorithms.

minimized for a given MDP realization in the offline problem. According to (12), the offline optimization problem can be written as follows:

$$\min_{\mathbf{X},\mathbf{B}} \sum_{n=0}^{N} \gamma^n \left[P_n \left(\delta(1 + X_n) L_n + \delta(X_n) R\right) + \delta(X_n) C_r + \delta(-X_n) C_d\right] \tag{22a}$$

$$\text{s.t.} \quad \frac{1}{\eta_d} \delta(-X_n) L_n + B_{\min} \le B_n, \tag{22b}$$

$$B_n \le B_{\max} - \eta_c \delta(X_n) R, \tag{22c}$$

$$B_{n+1} = B_n + \eta_c \delta(X_n) R - \frac{1}{\eta_d} \delta(-X_n) L_n, \tag{22d}$$

$$X_n \in \{0, 1\}, \quad n = 0, \ldots, N, \tag{22e}$$

where $\mathbf{X} = [X_0, X_1, \ldots, X_N]$ and $\mathbf{B} = [B_0, B_1, \ldots, B_N]$.

From definition (4), it can be seen that the function $\delta(x)$ is nonlinear. So the problem in (22a), (22b), (22c), (22d), and (22e) is a *nonlinear programming* (NLP) problem where the objective function or some of the constraints are nonlinear [27]. As we all know, it is difficult to solve the nonlinear optimization problem. For this reason, we will show that (22a), (22b), (22c), (22d), and (22e) can be mapped into a tractable linear programming before solving it.

Let us define the following variables regarding recharge and discharge operations in the slot t, respectively:

$$c_n = \begin{cases} 1, & \text{if recharging in the slot } n, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

$$d_n = \begin{cases} 1, & \text{if discharging in the slot } n, \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

where c_n indicates the recharge operation in the slot t, while d_n indicates the discharge operation in the slot t. Under the assumption that the recharge/discharge operations cannot be done simultaneously, there is a constraint on c_n and d_n in each slot as follows:

$$c_n + d_n \le 1. \tag{25}$$

Here, we define the vector $A_n = (c_n, d_n)$ as the joint decision variable to control the recharge/discharge operation in the

slot n. As a result, the optimization problem (22a), (22b), (22c), (22d), and (22e) can be rewritten as

$$\min_{\mathbf{A},\mathbf{B}} \quad \sum_{n=0}^{N} \gamma^n \left[P_n\left(\left(1-d_n\right)L_n + c_n R\right) + c_n C_r + d_n C_d\right] \quad (26a)$$

$$\text{s.t.} \quad c_n \eta_c R \le B_{\max} - B_n, \quad (26b)$$

$$d_n \frac{1}{\eta_d} L_n \le B_n - B_{\min}, \quad (26c)$$

$$B_{n+1} = B_n + c_n \eta_c R - d_n \frac{1}{\eta_d} L_n, \quad (26d)$$

$$c_n + d_n \le 1, \quad (26e)$$

$$c_n, d_n \in \{0,1\}, \quad n = 0, \ldots, N, \quad (26f)$$

where $\mathbf{A}$ is an optimal sequence of control decisions to (26a), (26b), (26c), (26d), (26e), and (26f), and $\mathbf{A} = [A_0, A_1, \ldots, A_N]$.

From (26a), (26b), (26c), (26d), (26e), and (26f), we can observe that the objective and constraint functions are linear. Moreover, the optimization variables c_n and d_n are constrained to be binary. Therefore, the problem in (26a), (26b), (26c), (26d), (26e), and (26f) is a *mixed integer linear programming* (MILP) problem. Currently, many existing tools can solve the MILP problem, such as GLPK [28], YALMIP [29], and *Ip_solve* [30]. In this paper, we employ *Ip_solve* to solve the proposed MILP problem. *Ip_solve* is a free linear (integer) programming solver based on the revised simplex method and the Branch-and-bound method for the integers, and it can solve pure linear, (mixed) integer/binary, semicontinuous and special ordered sets (SOS) models.

7. Performance Evaluation

In this section, the performance of the proposed dynamic energy storage control scheme is characterized quantitatively. Real-world workload traces and electricity price data sets are employed to evaluate the performance of the proposed scheme. In the following, we elaborate on the design of the experiments and presenting the experimental results.

7.1. Experimental Setup. In the experiments, we simulated a cloud-scale data center which hosts up to 2×10^4 servers [4]. For simplicity, we assume that the servers in data center are homogeneous, and it is easy to extend the experiments for the heterogeneous servers with little modifications. In order to evaluate the performance of the proposed scheme, we conducted experiments based on real-world workloads and electricity price data sets.

7.1.1. Workload Data. The real workload request is extracted from trace data gathered from Intel Netbatch Grid in 2012 [31]. We set the time slot size to 15 minutes, and count the number of job requests executed in each slot. The original traced period is only one month, so we repeat it for obtaining a three-month workload trace to complete the performance evaluation. Figure 2 shows the variations of workload requests in each 15 min period for four days. In order to perform the experiment with a larger-scale workload of data center, the number of requests extracted from Intel Netbatch Grid has to be scaled up. One of the approaches to scale up the workloads is to capture the underlying structure of the trace by separating the steady part and random part from the original workloads trace, and to scale the steady part up and add random part to it. However, it requires an appropriate method to capture the characteristics of the random part. We will try this approach to perform our experiment in the future work. In the current experiment, we assume that the number of users is scaled up by 1000 times, and accordingly, the number of requests should be statistically scaled up by 1000 times. The normal power consumption demand L_n for the workload request of data center in each slot n can be approximated by the following formula [4]:

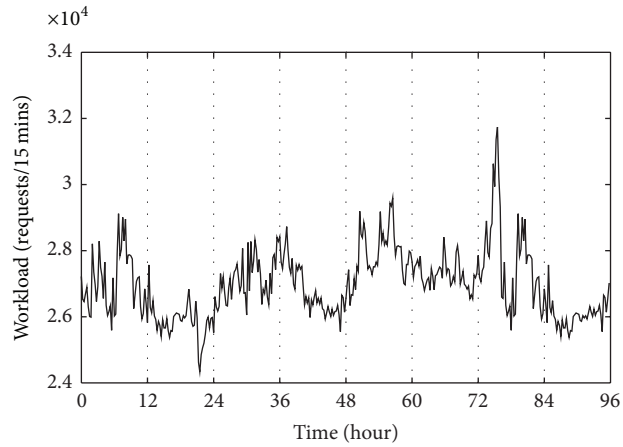

Figure 2: Workload arrival patterns of Intel Netbatch Grid for four days.

$$L_n = m_n \cdot \left(\alpha \mu_n^{\nu} + \beta\right) \cdot \text{PUE}, \quad (27)$$

where α, β, ν, and PUE are constants determined by the data center. Particularly, β is the average energy consumption of a server in one slot when it is idle, and μ_n denotes the number of workload requests served by one server in the slot n. Hence, $\alpha\mu_n^{\nu} + \beta$ gives the energy consumption of one server when it serves μ_n requests in one slot. m_n denotes the number of active servers in each slot n and has the maximum value $M = 2\times10^4$. PUE is the ratio of total power drawn by a data center facility (including cooling power) to IT equipment power. In today's energy-efficient data centers, the value of PUE is in interval from 1.1 to 2.0 generally, for example, Google data center has the average PUE of 1.12 in 2012 [32]. In our experiment, we set PUE = 1.2. The Intel Netbatch Grid is used for running its chip-simulation workloads, and it takes considerable time to serve one request by one server. According to the real workload trace, the average service time of each workload served by one server is 7-8 minutes, so we set $\mu_n = 2$. According to [4], $\alpha = 12.5$, $\beta = 150$ Watt, $\nu = 3$ when the CPU type of server in data center is AMD Athlon and service

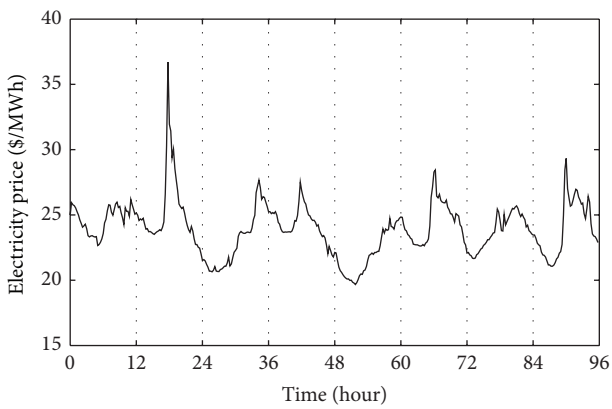

Figure 3: Real-time electricity prices at Houston from January 3 to January 6, 2013.

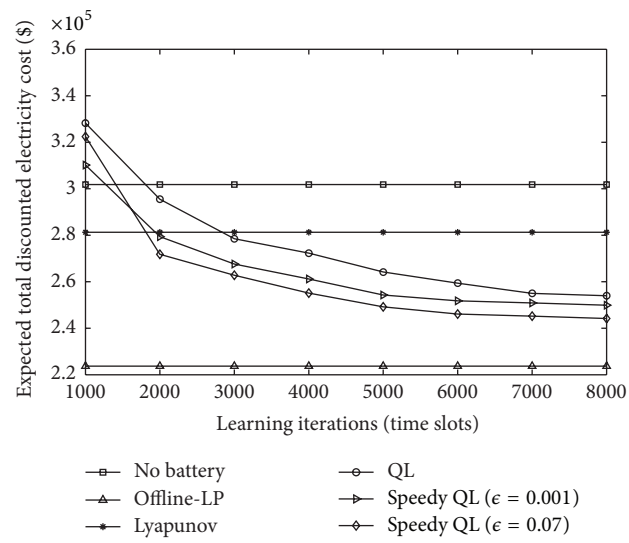

Figure 4: Expected total discounted electricity cost with respect to the number of iteration learning N_L.

rate is 2 requests/s. Accordingly, we set $\alpha = 11250$, $\beta = 13500$ Watt after calculation and $\nu = 3$ in our experiment.

7.1.2. Electricity Price Data. We use real-time electricity prices at Houston obtained from the Electric Reliability Council of Texas (ERCOT), and the real-time electricity prices vary on a 15 min basis [33]. The time horizon we consider in the experiment covers the period from January 1 to March 31, 2013. In these three months, there are 8640 real-time electricity price samples. Figure 3 shows the real-time electricity price variation characteristics at Houston from January 3 to January 6, 2013.

In the experiments, we simulated a time slotted system with slot duration of 15 minutes, that is, $m = 15$. The unit for energy consumption or battery energy level is MWh, and the unit for real-time electricity price is $/MWh. We discretize the energy consumption demand into 4 equal interval bins, with the boundaries specified by $\{[0, 950), [950, 1000), [1000, 1050), [1050, +\infty)\}$ MWh, and choose the energy consumption demand state space to be $\mathscr{L} = \{950, 1000, 1050, 1100\}$ MWh. Similarly, real-time electricity price is also discretized into 4 equal interval bins, with the boundaries specified by $\{[0, 20), [20, 30), [30, 40), [40, +\infty)\}$ $/MWh, and $\mathscr{C} = \{20, 25, 35, 40\}$ $/MWh is chosen to be as the electricity price state space. For a given maximum battery capacity $B_{\max}$, we also discretize the battery energy level into 4 equal interval bins, with the boundaries specified by $\{[0, 0.25B_{\max}), [0.25B_{\max}, 0.5B_{\max}), [0.5B_{\max}, 0.75B_{\max}), [0.75B_{\max}, B_{\max}]\}$ MWh, and choose the battery energy level state space to be $\mathscr{B} = \{0.25B_{\max}, 0.375B_{\max}, 0.625B_{\max}, 0.875B_{\max}\}$ MWh. Meanwhile, we let the power used for recharging the battery in one slot is 500 MWh, that is, $R = 500$ MWh, and let the discount factor $\gamma = 0.9$, which has been justified in [34]. Since the minimum energy level $B_{\min}$ is a constant, the value of $B_{\min}$ has no effect on the experimental results, and we set $B_{\min} = 0$ in the experiments.

7.2. Experimental Results. In order to demonstrate the performance improvement of the proposed dynamic energy storage control algorithm, we considered the Lyapunov optimization algorithm [19] and the offline optimization problem. The Lyapunov optimization algorithm makes decisions to recharge/discharge the battery for minimizing the electricity cost using the solution with threshold structure. The solution of the offline optimization problem can be considered as an lower bound on the performance of the proposed learning theoretic algorithm and the Lyapunov optimization algorithm.

7.2.1. Impact of the Number of Learning Iteration. In the first experiment, we intend to investigate the convergence rate and performance improvement of the proposed scheme using the real-world workload and electricity price traces. Let N_L denote the number of learning iterations. The value for N_L covers $\{1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000\}$, and the battery maximum capacity $B_{\max}$ is chosen to be 5000 MWh. The initial battery energy level is set to be zero, that is, $B_0 = 0$, and we evaluate the optimal policy for a fully efficient battery ($\eta_c = \eta_d = 1$). The new battery costs involve a unit price g_b (in $ per MWh) [35]. That is, for a new battery with the capacity B, the battery cost is given by $C_b = g_b B$. Here, we set g_b to 100 $/MWh, and the recharge/discharge cycles K to 2800. The other parameters for the Lyapunov optimization are set as follows: the constant $\chi_{\min} = C_{\max}$, where $C_{\max}$ denotes the maximum of the real-time electricity price, the control parameter $V = V_{\max}$, and the maximum power that can be drawn from the grid in any slot $P_{\text{peak}} = L_{\max} + R$, where $L_{\max}$ is the maximum energy consumption demand in any slot. The parameters set above for Lyapunov optimization are justified in [19].

In Figure 4, we illustrate the expected total electricity cost by the Q-Learning based approaches against the number of learning iteration, N_L, together with the performance of the Lyapunov optimization approach. As shown in Figure 4, it can be observed that for $N_L \geq 3000$ the Q-Learning based

approaches have less total electricity cost than Lyapunov optimization, and for $N_L \geq 5000$ the Speedy Q-Learning algorithm with $\epsilon = 0.07$ yields approximately 11% more electricity cost than the offline solution. We also see that the expected total electricity costs for the Q-Learning based approaches decrease as the number of iteration learning N_L increases, while the costs for the Lyapunov optimization and the data center without energy storage facilities do not vary with N_L. The reason for the trend of the total costs of Q-Learning based approaches with N_L is that the larger N_L implies that more accurate $Q^{\pi^*}(s,x)$ is estimated, thus the policy taken by estimated $Q^{\pi^*}(s,x)$ is closer to the optimal policy, so the lower cost is yielded. The result shows that the Q-Learning based approaches can approximate the optimal policy with increasing accuracy as N_L increases. From the Figure 4, it can also be observed that for Speedy Q-Learning algorithm with a low exploration probability ($\epsilon = 0.001$) causes low learning rate, compared to the exploration probability ($\epsilon = 0.07$). Speedy Q-Learning algorithm with $\epsilon = 0.07$ has faster convergence rate of $Q_n(s,x)$ to $Q^{\pi^*}(s,x)$ than the standard Q-Learning ($\epsilon = 0.07$). This is because that the speedy Q-Learning algorithm uses two successive estimates of the state-action value function to update the Q-values in order to achieve faster convergence. Since larger ϵ is more likely to explore better action which might remain unexplored, it can accelerate convergence of $Q_n(s,x)$ to $Q^{\pi^*}(s,x)$. Therefore, with a suitable choice of $\epsilon > 0$, Speedy Q-Learning algorithm may be able to strike a balance in the exploration versus exploitation trade off, and achieve a faster convergence rate.

Figure 5 shows the long-run average electricity cost for different number of learning iteration N_L. It can be observed from Figure 5 that the long-run average electricity costs of the Q-Learning based approaches decrease as the number of learning iteration N_L, while the costs for the Lyapunov optimization and the data center without energy storage facilities remain unchanged as N_L varies. For $N_L \geq 4000$, the Speedy Q-Learning algorithm ($\epsilon = 0.001$ or $\epsilon = 0.07$) yields lower average cost than the Lyapunov optimization algorithm. Compared with the Speedy Q-Learning algorithm with $\epsilon = 0.001$ and standard Q-Learning algorithm, the Speedy Q-Learning algorithm with $\epsilon = 0.07$ has better performance. The reason is that for smaller number of learning iteration N_L, the error between the Q-value estimated by Q-Learning based algorithm and the optimal Q-values is larger, then the policies are not optimal and this results in higher average costs. As the number of learning iteration increases, more accurate Q-values are estimated and Speedy Q-Learning with larger ϵ also accelerates convergence of Q-values, and then more cost can be saved.

7.2.2. Impact of Battery Capacity. In this subsection, we further carried out an experiment in order to investigate the impact of the battery capacities of data centers by setting $B_{\max} = 2000, 3000, 4000, 5000, 6000$ MWh. We chose the number of learning iteration $N_L = 6000$, the exploration probability $\epsilon = 0.07$. The other parameters and the simulation settings were the same as those in Section 7.2.1.

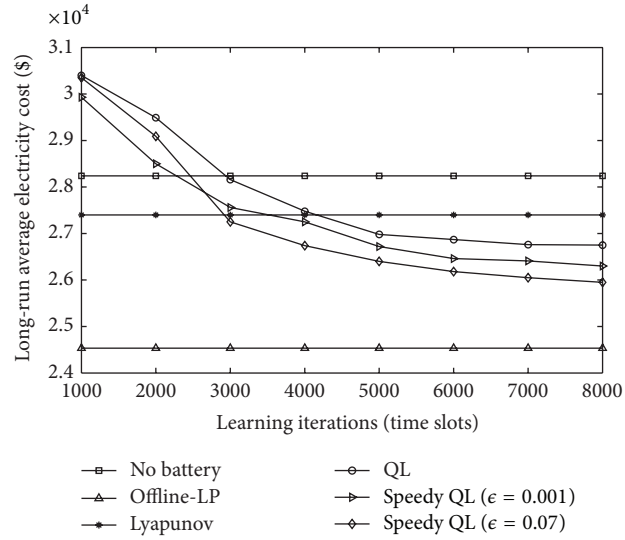

Figure 5: Long-run average electricity cost with respect to the number of iteration learning N_L.

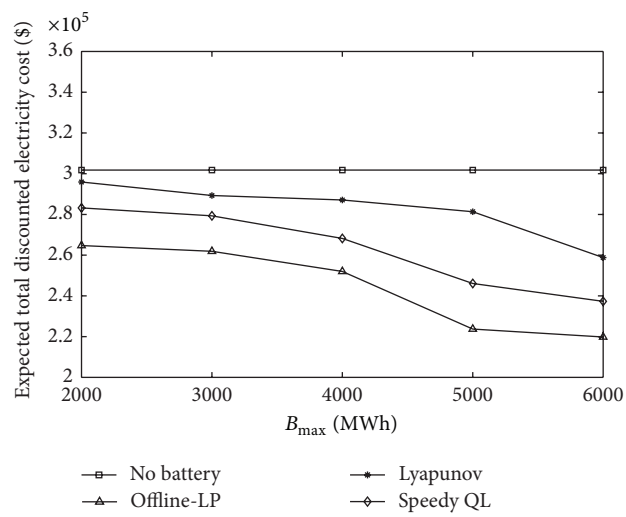

Figure 6: Expected total discounted electricity cost for different $B_{\max}$ and $N_L = 6000$.

In Figure 6 we show the impact of battery capacity, $B_{\max}$, on the expected total electricity cost for $N_L = 6000$. It can be observed that the expected total electricity cost decreases upon increasing $B_{\max}$, that is, the larger the battery capacity is, the more cost saving by the Speedy Q-Learning based scheme can be obtained. Additionally, we also see that the Speedy Q-Learning algorithm with $\epsilon = 0.07$ yields at most approximately 10% more electricity cost than the offline solution, and lower than the Lyapunov optimization algorithm. The reason is that for larger battery capacity the Speedy Q-Learning based scheme would be likely to make the optimal policy to store more power at lower prices, while the threshold structure of the optimal solution for the Lyapunov optimization algorithm has no capability of learning system dynamics, and it stores power at the prices lower than

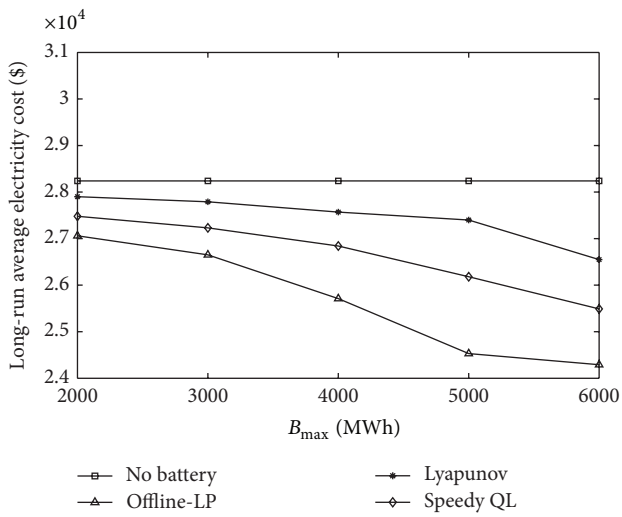

Figure 7: Long-run average electricity cost for different $B_{\max}$ and $N_L = 6000$.

the thresholds, but higher than the prices used by the Speedy Q-Learning based scheme.

Figure 7 shows the long-run average electricity costs for different battery capacities $B_{\max}$. We plot the performance of the Speedy Q-Learning based scheme for $N_L = 6000$ and $\epsilon = 0.07$, compared with the other approaches. From Figure 7, it can be observed that as $B_{\max}$ increases, the average costs yielded by Speedy Q-Learning algorithm and Lyapunov optimization algorithm decrease, while the Speedy Q-Learning algorithm achieves more average cost saving than the Lyapunov optimization algorithm.

8. Conclusion

In this paper, we investigated the problem of electricity cost minimization of data centers using energy storage for time-varying electricity prices under deregulated electricity markets, which was formulated as a discounted cost Markov decision process. A dynamic energy storage control strategy based on the Q-Learning algorithm was designed to reduce the electricity cost, and we also applied the Speedy Q-Learning algorithm in order to accelerate convergence. The advantage of the proposed scheme is that it makes decision without any priori information about the energy management system of the data centers, and it can also adapt to the variations of the workload and the electricity prices. We also studied the offline optimization problem which was characterized as an MILP problem, and its optimal solution can be considered as a lower bound on the performance of the proposed algorithm. In the experiments, real workload traces and electricity price data sets were used for verifying the performance of the proposed scheme. The results illustrated the effectiveness of the proposed scheme in saving the electricity cost via comparison with the benchmark algorithm. Results for the real traces that may not provably follow the Markovian assumption also show that the proposed scheme generally performs well.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National "863" Project of China (no. 2012AA050802), the Fundamental Research Funds for the Central Universities (WK2100100021), and National Natural Science Foundation of China (no. 61174062).

References

[1] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 123–134, 2009.

[2] J. Koomey, *Growth in Data Center Electricity Use*, Analytics Press, Burlingame, Calif, USA, 2011.

[3] G. Ghatikar, V. Ganti, N. Matson, and M. A. Piette, "Demand response opportunities and enabling technologies for data centers: findings from field studies," Tech. Rep. LBNL-5763E, 2012, Lawrence Berkeley National Laboratory, Berkeley, Calif, USA, 2012.

[4] Y. Guo and Y. Fang, "Electricity cost saving strategy in data centers by using energy storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1149–1160, 2013.

[5] M. Lin, *Algorithmic issues in green data centers [M.S. thesis]*, California Institute of Technology, 2011.

[6] B. Fortenbery and W. Tschudi, "DC power for improved data center efficiency," Tech. Rep., Lawrence Berkeley National Laboratory, Berkeley, Calif, USA, 2008.

[7] Y. Liu, R. Guan, J. Ma, and K. Zhang, "Research of data center fresh air ventilation cooling system," in *Proceedings of the 8th International Symposium on Heating, Ventilation and Air Conditioning (ISHVAC '14)*, pp. 299–306, 2014.

[8] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '09)*, pp. 145–150, August 2009.

[9] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[10] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.

[11] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, "Energy efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," in *Proceedings of the 8th International Symposium on High-Performance Computer Architecture (HPCA '02)*, pp. 29–40, 2002.

[12] A. Alimonda, S. Carta, A. Acquaviva, A. Pisano, and L. Benini, "A feedback-based approach to DVFS in data-flow applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 11, pp. 1691–1704, 2009.

[13] C. Xian, Y.-H. Lu, and Z. Li, "Dynamic voltage scaling for multitasking real-time systems with uncertain execution time,"

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 8, pp. 1467–1478, 2008.

[14] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, 2000.

[15] J. Yang, K. Zeng, H. Hu, and H. Xi, "Dynamic cluster reconfiguration for energy conservation in computation intensive service," *IEEE Transactions on Computers*, vol. 61, no. 10, pp. 1401–1416, 2012.

[16] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid '10)*, pp. 826–831, Melbourne, Australia, May 2010.

[17] L. Rao, X. Liu, L. Xie, and W. Liu, "Coordinated energy cost management of distributed internet data centers in smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 50–58, 2012.

[18] J. Luo, L. Rao, and X. Liu, "Data center energy cost minimization: a spatio-temporal scheduling approach," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 340–344, Turin, Italy, April 2013.

[19] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramanian, "Optimal power cost management using stored energy in data centers," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '11)*, pp. 221–232, San Jose, Calif, USA, June 2011.

[20] W. Shi and V. W. S. Wong, "Real-time vehicle-to-grid control algorithm under price uncertainty," in *Proceedings of the IEEE 2nd International Conference on Smart Grid Communications (SmartGridComm '11)*, pp. 261–266, October 2011.

[21] P. M. van de Ven, N. Hegde, L. Massoulié, and T. Salonidis, "Optimal control of end-user energy storage," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 789–797, 2013.

[22] R. Bellman, "A Markovian decision process," *Journal of Mathematical Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.

[24] T. Bach, J. Yao, and S. Yang, "Fuzzy Q-learning for uniform price wholesale power markets," in *Proceedings of the 3rd International Conference on Communication Systems and Network Technologies (CSNT '13)*, pp. 578–582, Gwalior, India, April 2013.

[25] C. Watkins, *Learning from delayed rewards [Ph.D. thesis]*, Department of Computer Science, Cambridge University, Cambridge, UK, 1989.

[26] M. G. Azar, R. Munos, M. Ghavamzadeh, and H. Kappen, "Speedy Q-learning," in *Advances in Neural Information Processing Systems*, 2011.

[27] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," in *50 Years of Integer Programming 1958–2008*, M. Jünger, Ed., pp. 561–618, 2010.

[28] GLPK, http://www.gnu.org/software/glpk/.

[29] YALMIP, http://users.isy.liu.se/johanl/yalmip/.

[30] Ip solve, http://lpsolve.sourceforge.net/.

[31] O. Shai, E. Shmueli, and D. G. Feitelson, "Heuristics for resource matching in intel's compute farm," in *Proceedings of the 17th Workshop on Job Scheduling Strategies for Parallel Processing, in Conjunction with IPDPS*, May 2013.

[32] Google Data Center PUE, http://www.google.cn/about/datacenters/.

[33] The Electric Reliability Council of Texas (ERCOT), http://www.ercot.com/.

[34] K. J. Prabuchandran, S. K. Meena, and S. Bhatnagar, "Q-learning based energy management policies for a single sensor node with finite buffer," *IEEE Wireless Communications Letters*, vol. 2, no. 1, pp. 82–85, 2013.

[35] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar, "Using batteries to reduce the power costs of internet-scale distributed networks," in *Proceedings of the 3rd ACM Symposium on Cloud Computing (SoCC '12)*, San Jose, Calif, USA, October 2012.