An Online Algorithm for Power-proportional Data Centers with Switching Cost

Ming Zhang[1], Zizhan Zheng[2] and Ness B. Shroff[3]

Abstract— **Recent studies have shown that power-proportional data centers can save energy cost by dynamically "right-sizing" the data centers based on real-time workload. More servers are activated when the workload increases while some servers can be put into the sleep mode during periods of low load. In this paper, we revisit the dynamic right-sizing problem for heterogeneous data centers with various operational cost and switching cost. We propose a new online algorithm based on a regularization technique, which achieves a better competitive ratio compared to the state-of-the-art greedy algorithm in [17]. We further introduce a switching cost offset into the model and extend our algorithm to this new setting. Simulations based on real workload and renewable energy traces show that our algorithms outperform the greedy algorithm in both settings.**

I. Introduction

Internet-scale services like web-mail, live streaming, online gaming and social networks usually have millions or even billions of active users everyday. Providers like Google, Amazon and Facebook, in order to maintain the reliability, accessibility and guaranteed performance of their systems, have deployed numerous large data centers including massive number of servers, causing a huge amount of electricity and cooling cost. Based on [16], the electricity consumption of large data centers has accounted for 1.3% of all the electricity use of the world and almost 2% of the United States in 2010.

Recent research [20]–[22] shows that the energy cost can be significantly reduced by dynamically distributing the workload to various data centers based on the idea of "Geographical Load Balancing" (GLB) and "Right-sizing" to make the data center more power-proportional [4], [8], [18]. Specifically, the central load balancer dynamically dispatches the workload requests to geographically located data centers that consist of thousands of servers. Each data center dynamically adjusts the number of active servers to serve the requests so that during low-load period, servers that do not have jobs transfer to the power-saving mode or are shut down completely after data and operation states are reserved.

In [17], [18], Lin et al., propose a cost minimizing model for the "right-sizing" of data centers incurring both operational cost and switching cost. Their model is a general convex optimization problem where the objective function consists of two parts representing the operational cost and switching cost, respectively. Examples are given to show how concrete energy and delay costs of data centers can fit into their model. The operational cost is modeled as a time-dependent convex function and a linear function is used to represent the switching cost of switching certain number of servers from power-saving mode to active mode to serve the increasing workload [6]. The switching cost is incurred only when the number of active servers increase. Such switching cost not only includes the total energy cost, but also delay in data migration, increased wear-and-tear on servers and the risk involved in server mode toggling. A 3-competitive online algorithm [18] is proposed for the case of a single server. In [17], Lin et al. consider a more general setting with multiple heterogeneous servers and look-ahead information, and propose the AFHC online algorithm that is $(1+\frac{\beta}{e_0})$-competitive where β is the maximum value of unit switching cost and e_0 refers to the minimum unit operational cost of all data centers. The algorithm reduces to the simple greedy algorithm when the look-ahead window size is zero. The two online algorithms are the first attempts to deal with the online convex right-sizing problem with switching cost and provide performance guarantees. However, the 3-competitive algorithm only works for the single-server setting, while the greedy algorithm can have a large competitive ratio when the minimum value of the unit operating cost e_0 is very small compared to the switching cost.

In this paper, we revisit the right-sizing problem studied in [17] and propose a better algorithm. We consider a system with multiple data centers located in different places. The operational cost and switching cost of each data center vary based on the local energy prices, the availability of renewable energy, and other factors such as energy storage and servers' wear-and-tear cost. The information of the workload and cost functions of each data center are both revealed only at the beginning of each time slot. There is no look-ahead window, meaning that information for future time slots is not available at the central load balancer. Real-time workload demand is dispatched to different data centers at each time slot by the central load balancer which tries to minimize the total cost for all time slots.

We develop a new online algorithm based on the regularization technique proposed in [7] for the right-sizing problem. We show that our online regularization algorithm achieves a better competitive ratio compared to the greedy algorithm in [17]. We further extend our system model by introducing a new time-dependent parameter, called switching cost offset, which enables a data center to serve the increase in workload demand without incurring any switching cost when the increase is less than the offset parameter. This new parameter is meaningful since each data center may have access to some local renewable energy source or energy

[1] Ming Zhang is Ph.D. student with the Department of CSE, The Ohio State University, U.S.A.

[2] Zizhan Zheng is with Faculty of the Department of Computer Science, Tulane University, U.S.A.

[3] Ness B. Shroff is with Faculty of the Department of ECE and CSE, The Ohio State University, U.S.A.

storage, which allows it to activate some number of sleeping servers by paying a negligible cost. In addition, the switching cost offset also includes the delay tolerance during data migration when servers are activated or turned down and the cost compensation by certain protection mechanism to reduce the server state toggling cost. To the best of our knowledge, this is the first work that considers such a switching cost offset. We propose another online regularization algorithm with guaranteed performance.

Our main contribution can be summarized as follows:

1) We propose an online regularization algorithm for the right-sizing problem of multiple heterogenous data centers with various operational cost and switching cost. We prove a competitive ratio of our algorithm in terms of the switching cost and the operational cost functions, which is always better than that of the greedy algorithm in [17].
2) We consider an extension of the right-sizing problem by introducing a switching cost offset into the model, and propose an online algorithm with guaranteed performance. Our algorithm is the first attempt to deal with this important extension.

The rest of the paper is organized as follow: Section II discusses the related work on energy cost minimization of data centers and the regularization method. Section III introduces our general system model and the online algorithm is proposed in Section IV. We then discuss the model with switching cost offset and the corresponding algorithm in Section V. Numerical results are given in Section VI and we conclude the paper in Section VII.

II. Related Work

There are multiple recent works [8], [10], [12]–[14], [20]–[22] discussing the "Geographical Load Balancing" and the "Right-sizing" problem for data centers. The most relevant works are [17], [18]. In [18], Lin et al., considered the case of a single data center where the data center determines the workload (number of active servers) based on general convex operational cost function and linear switching cost. They proposed the "Laze Capacity Provisioning" online algorithm by utilizing the structure of the optimal offline solution, which achieves a competitive ratio of 3. Later, Bansal et al. [5] improved the competitive ratio to 2 by proposing a new randomized online algorithm. However, the online algorithms in both [18] and [5] only work for the single data center case and their performance can be arbitrarily bad for the case with multiple heterogenous data centers. In [17], Lin et al., considered the "right-sizing" problem for the heterogeneous data center model and proposed the "Averaging Fixed Horizon Control" algorithm. They proved that their algorithms achieves a competitive ratio that depends on the switching cost and the convex operational cost function. In this work, we revisit the heterogeneous case and propose a new online algorithm with a better performance guarantee.

There are extensive studies on online algorithm design for cloud resource management [15], [24] and real-time dispatch [9], [28], [29]. More specifically, [7] first introduced the concept of regularization for online algorithm design and proved that the online algorithm with a regularization term can achieve a competitive ratio proportional to $\log N$ where N is number of variables. In [27], Zhang et al. investigate online resource management for Cloud-based content delivery networks. They proposed an efficient online algorithm by using the regularization technique and proved its performance guarantee. However, the cost function is linear in all these works and none of them take the switching cost offset into consideration. In this work, we consider the more general and practical case of convex operational cost and compare the performance of regularization based online algorithms and the greedy algorithms.

III. System Model

In this section, we discuss our system model of multiple heterogeneous data centers located in various places and the overall optimization problem.

We study a system consisting of a central load balancer and N heterogeneous data centers, each with thousands of servers located in different places. The servers in each data center are assumed to be homogeneous as in previous work. The central load balancer distributes the workload to data centers and each data center either activates or deactivates a certain number of servers to serve the workload. The cost of each data center for serving the workload consists of two parts, operational cost and switching cost, both may vary across data centers.

In this paper, we consider general classes of cost functions. The operational cost is modelled by a time-dependent non-decreasing convex function $f_{i,t}(\cdot)$, where $f_{i,t}(s_i(t))$ refers to the total operational cost for data center i with workload $s_i(t)$ at time t, which includes the energy cost for serving the workload as well as the cost associated with data transmission and delay, etc. We assume that $f_{i,t}(\cdot)$ is continuously differentiable. In [18], Lin et al. provided concrete examples to show how the real data center cost can be fitted into this general convex operational cost model.

In addition to the operational cost, data centers incur a switching cost for activating servers, which is modelled by a one-directional linear function. The switching cost generally includes the energy cost of transferring server states, data migration latency, server state toggling risk, and the wear-and-tear cost [18]. Further, when the number of active servers and computing resources are reduced, the user experience may be degraded, resulting a decline in revenue [19], which can also be captured by the switching cost. Formally, when the workload dispatched to data center i changes from $s_i(t-1)$ to $s_i(t)$, it incurs a switching cost $\beta_i(s_i(t) - s_i(t-1))^+$ where the function $(x)^+ = \max\{0, x\}$ and β_i is a coefficient. We only take into consideration the switching cost when the workload increases, since turning off servers usually has a negligible cost [18], [23].

We further extend the model to the situation where each data center has access to local renewable energy, or has a certain protection mechanism and delay tolerance. For instance, protection mechanisms can reduce the wear-and-tear cost and the corresponding risk involved in the state toggling of servers [11], while delay tolerant workload is less sensitive to the latency for toggling servers out of power-saving mode. On the other hand, timing-varying renewable energy supply can help reduce both the operational and switching costs.[1] To model these effects, we introduce a time-dependent offset parameter $r_i(t)$ into our model such that no switching cost is incurred when the increase of workload is less than or equal to $r_i(t)$. Such an offset parameter allows each data center to activate a certain number of servers without incurring any switching cost. With this offset incorporated, the switching cost for data center i when its workload changes from $s_i(t-1)$ to $s_i(t)$ now becomes $\beta_i(s_i(t)-s_i(t-1)-r_i(t))^+$.

We consider a time-slotted system from $t = 1$ to $t = T$, and an online setting where all the future information including the operational costs, switching costs, and the workload is unknown. The central load balancer is only aware of all the parameters at the current and past time slots. Our objective is to minimize the overall cost by dynamically dispatching the workload to each data center at the beginning of each time slot under the constraint that the total demand must be satisfied at each time slot.

At the beginning of each time-slot t, the workload $D(t)$, operational cost function $f_{i,t}(\cdot)$, and the offset $r_i(t)$ are revealed. The central load balancer then distributes workload $s_i(t)$ to data center i, incurring a operational cost $\sum_{i=1}^N f_{i,t}(s_i(t))$ and a switching cost $\sum_{i=1}^N \beta_i(s_i(t)-s_i(t-1) - r_i(t))^+$ at time slot t. Note that we do not explicitly include a capacity constraint for each data center in (1). This can be easily modeled by setting the operational cost $f_{i,t}$ to infinity when the workload assigned to data center i exceeds its capacity. As long as $f_{i,t}(s)$ is continuously differentiable when s is within the capacity region, all the results in this paper remain valid. The objective of the central load balancer is to minimize the overall operational cost and switching cost among all time slots as in (1).

$$\begin{aligned}
\min_{s_i(t)} \sum_{t=1}^{T} \sum_{i=1}^{N} & \left[f_{i,t}(s_i(t)) + \beta_i(s_i(t) - s_i(t-1) - r_i(t))^+\right] \\
s.t. \; & \sum_{i=1}^{N} s_i(t) \geq D(t) \; \forall t \\
& s_i(t) \geq 0 \; \forall i, t \\
& s_i(0) = 0 \; \forall i
\end{aligned} \tag{1}$$

We use competitive ratio as the performance metric throughout this paper. Denote $A(1:t)$ as the input information (e.g., the workload $D(t)$, the operational cost function $f_{i,t}(\cdot)$ and the switching cost offset $r_i(t)$) from time slot 1 to time slot t. For an online algorithm π, the decision $s_i(t)$ $\forall i$ at each time slot t can only be based on input $A(1:t)$. Let $C_\pi(A(1:T))$ be the total cost of algorithm π and we compare it with the total cost of the optimal offline solution $C_{opt}(A(1:T))$ which is obtained by solving (1). Then, the competitive ratio of algorithm π is given by

$$CR_\pi = \max_{A(1:T)} \frac{C_\pi(A(1:T))}{C_{opt}(A(1:T))}$$

Table I summarizes the notations used in the paper.

TABLE I

LIST OF NOTATIONS

Symbol	Meaning
T	Number of time slots
N	Number of data centers
$D(t)$	Total workload demand at time t
D_{max}	$\max_t D(t)$
$f_{i,t}(\cdot)$	Operational cost of data center i at time t
β_i	Coefficient of switching cost for data center i
β	$\max_i \beta_i$
$r_i(t)$	Switching cost offset for data center i at time t
$s_i(t)$	Workload dispatched to data center i at time t

IV. Online Regularization Algorithm

In this section, we study the right-sizing problem without switching offset, that is $r_i(t) = 0$ $\forall i, t$. We first review the greedy algorithm (a.k.a the AFHC algorithm in [17] without look-ahead information) and its performance guarantee. Then, we present our online regularization based algorithm and compare the competitive ratios of the two algorithms.

A. The Greedy Algorithm

Lin et al. [17] proposed the AFHC algorithm and analyzed the pros and cons of AFHC compared to the classic Receding Horizon Control (RHC) algorithm. They claimed that the AFHC algorithm can outperform the RHC algorithm when there are multiple heterogeneous data centers. Both AFHC and RHC work for the case with look-ahead information. When there is no look-ahead information as we consider in the paper, both algorithms reduce to the simple greedy algorithm. That is, both algorithm compute the load assignment $\tilde{s}_i(t)$ by solving the following optimization problem in each time-slot.

$$\begin{aligned}
\tilde{s}_i(t) = \text{argmin}_{s_i(t)} & \sum_{i=1}^{N} \left[f_{i,t}(s_i(t)) + \beta_i(s_i(t) - \tilde{s}_i(t-1))^+\right] \\
s.t. \; & \sum_{i=1}^{N} s_i(t) \geq D(t) \; \forall t \\
& s_i(t) \geq 0 \; \forall i, t
\end{aligned} \tag{2}$$

Let $e_{0,i}$ denote the minimum positive constant such that $f_{i,t}(x) \geq e_{0,i}x$, $\forall x, t$. The following result is proved in [17]:

Theorem IV.1. *The greedy algorithm is* $(1 + \beta/e_0)$*-competitive where* $\beta = \max_i\{\beta_i\}$ *and* $e_0 = \min_i\{e_{0,i}\}$.

We note that since $f_{i,t}(x)/x$ may approach to 0 when x is close to 0, e.g. when $f_{i,t}(x) = x^\alpha$ for $\alpha > 1$, the value of β/e_0 can be huge. Moreover, our simulation results using real data from Google Cloud platform (see Section VI) indicate that the greedy algorithm may cause unnecessary frequent server switching, leading to bad performance. To tackle these issues, we present a new online algorithm based on a regularization technique for problem (1) in the following subsection, which achieves a better competitive ratio and shows better empirical performance.

[1] In this work, we assume that the allocation of renewable energy for reducing operating cost and that for reducing switching cost follows a predetermined scheme, and incorporate the former into the operational cost function.

Algorithm 1 Online Regularization with Convex Operational Cost

```
Input: $\epsilon > 0$ and $\eta = \ln(1 + ND_{max}/\epsilon)$
Initialization: $\tilde{s}_i(0) = 0$ for all $i = 1, \cdots, N$
for $t = 1$ to $T$ do
    $\tilde{s}_i(t) = \text{argmin}_{s_i(t) \in P_t} \Big\{ f_{i,t}(s_i(t)) + \frac{1}{\eta} \sum_{i=1}^{N} \beta_i \Big[ (s_i(t) + \epsilon/N) \ln \Big( \frac{s_i(t) + \epsilon/N}{\tilde{s}_i(t-1) + \epsilon/N} \Big) - s_i(t) \Big] \Big\}$    (3)
    where $P_t \triangleq \{ s_i(t) | \sum_{i=1}^{T} s_i(t) \geq D(t), s_i(t) \geq 0 \ \forall i \}$
end for
```

B. Online Regularization Algorithm

Our algorithm adopts the novel framework proposed in [7] for designing competitive online algorithms. The algorithm is essentially greedy by solving a convex optimization problem in each round, where the objective function includes both the operating cost and the regularized switching cost. As in [7], we use the relative entropy plus a linear term as the regularizer. The regularizer for two (discrete) distributions $\boldsymbol{\theta}$ and $\boldsymbol{u}$ is defined as $\sum_i \theta_i \ln(\theta_i/u_i) + \theta_i - u_i$. But unlike the regularization algorithm in [7] where the operational cost function is linear and all the variables are within the range of $[0, 1]$, our algorithm deals with convex functions and general non-negative domains for all the variables $s_i(t)$. We present our online algorithm in Algorithm 1.

In Algorithm 1, we assume that $D_{max} = \max_t D(t)$ is known in advance and N refers to the number of data centers. We compute the workload dispatch $\tilde{s}_i(t)$ by solving the convex optimization problem (3) in each time-slot, where ϵ is a parameter that can be adjusted. Since (3) is a continuous convex optimization problem, it can be solved in polynomial time. Algorithm 1 computes $\tilde{s}_i(t)$ using only the information available at the current time-slot and $\tilde{s}_i(t-1)$.

To study the performance of our online algorithm, we adopt a primal-dual analysis similar to [7]. Below we first provide an overview of the main idea of the primal-dual technique involved in the analysis. We start with the primal problem (1) with $r_i(t) = 0 \ \forall i, t$, which is equivalent to the following where we introduce variables $z_i(t)$ so that the objective function is continuous:

$$
\begin{aligned}
\min_{s_i(t)} & \sum_{t=1}^{T} \sum_{i=1}^{N} \left(f_{i,t}(s_i(t)) + \beta_i z_i(t) \right) \\
s.t. \ & \sum_{i=1}^{N} s_i(t) \geq D(t) \ \forall t \\
& s_i(t) \geq 0 \ \forall i, t \\
& z_i(t) \geq s_i(t) - s_i(t-1) \ \forall i, t \\
& z_i(t) \geq 0 \ \forall i, t
\end{aligned}
\tag{4}
$$

The Lagrangian function of (4) is

$$
\begin{aligned}
& L(\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}, s_i(t), z_i(t)) \\
&= \sum_{t=1}^{T} \sum_{i=1}^{N} [f_{i,t}(s_i(t)) + \beta_i z_i(t)] + \sum_t \lambda_t [D(t) - \sum_i s_i(t)] \\
&+ \sum_t \sum_i \mu_{i,t} \left(s_i(t) - s_i(t-1) - z_i(t) \right) \\
&- \sum_t \sum_i \left(l_{i,t} s_i(t) - k_{i,t} z_i(t) \right) \\
&= \sum_t \sum_i [f_{i,t}(s_i(t)) + (\mu_{i,t} - \mu_{i,t+1} - \lambda_t - l_{i,t}) s_i(t)] \\
&+ \sum_t \sum_i \left(\beta_i - \mu_{i,t} - k_{i,t} \right) z_i(t) + \sum_t \lambda_t D(t)
\end{aligned}
$$

where λ_t, $l_{i,t}$, $\mu_{i,t}$ and $k_{i,t}$ are the Lagrangian multipliers for the four constraints in (4). Thus, the dual function of (4) is

$$
\begin{aligned}
& D(\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}) = \min_{s_i(t), z_i(t)} L(\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}, s_i(t), z_i(t)) \\
&= \min_{s_i(t)} \sum_t \sum_i [f_{i,t}(s_i(t)) + (\mu_{i,t} - \mu_{i,t+1} - \lambda_t - l_{i,t}) s_i(t)] \\
&+ \min_{z_i(t)} \sum_t \sum_i \left(\beta_i - \mu_{i,t} - k_{i,t} \right) z_i(t) + \sum_t \lambda_t D_t
\end{aligned}
\tag{5}
$$

To establish a relation between the optimal offline solution and the online solution, the main idea is to assign the dual variables with values $\hat{\mu}_{i,t}$, $\hat{\lambda}_t$ $\hat{l}_{i,t}$ and $\hat{k}_{i,t}$ based on the optimal online solution $\tilde{s}_i(t)$. The weak duality tells us

$$\max_{\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}} D(\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}) \leq \text{value of (1)}$$

Therefore, if we can prove

$$\text{Total Online Cost} \leq \Lambda \cdot D(\hat{\mu}_{i,t}, \hat{\lambda}_t, \hat{l}_{i,t}, \hat{k}_{i,t})$$

for some $\Lambda > 1$, then our online algorithm is Λ-compeitive, that is,

$$\text{Total Online Cost} \leq \Lambda \cdot \text{value of (1)}$$

All the theorems in this section and Section V are based on this idea. We first show that Algorithm 1 has a smaller competitive ratio compared with the greedy algorithm.

Theorem IV.2. *Algorithm 1 is* $(1 + \frac{\beta}{e_0 + C})$*-competitive where*

$$C \triangleq \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \frac{\beta_i}{\eta} \ln \left(\frac{\tilde{s}_i(t) + \epsilon/N}{\tilde{s}_i(t-1) + \epsilon/N} \right) \tilde{s}_i(t)}{\sum_{t=1}^{T} D(t)} \tag{6}$$

and $C \in [0, \beta]$.

Proof. To show $C \in [0, \beta]$, we need two inequality facts: $\sum_i a_i \log(a_i/b_i) \geq (\sum_i a_i) \log(\frac{\sum_i a_i}{\sum_i b_i}) \quad \forall a_i, b_i > 0$ and $a \ln(a/b) \geq a - b \ \ \forall a, b > 0$. Due to limited space, we omit the details of this part. Please refer to [26] for a proof.

Next, we show that the competitive ratio of Algorithm 1 is $1 + \frac{\beta}{e_0 + C}$. We assign dual variables $\tilde{\lambda}_t$ and $\tilde{l}_{i,t}$ to the constraints $\sum_i s_i(t) \geq D(t) \ \forall t$ and $s_i(t) \geq 0 \ \forall i, t$ in (3) respectively. Since (3) is a convex optimization problem, by applying the KKT conditions of (3), we have for any i and t,

$$\frac{\beta_i}{\eta} \ln \left(\frac{\tilde{s}_i(t) + \epsilon/N}{\tilde{s}_i(t-1) + \epsilon/N} \right) = \tilde{\lambda}_t - f'_{i,t}(\tilde{s}_i(t)) + \tilde{l}_{i,t} \tag{7}$$

By setting $\lambda_t = \tilde{\lambda}_t$, $\mu_{i,t} = \frac{\beta_i}{\eta} \ln \left(\frac{D_{max} + \epsilon/N}{\tilde{s}_i(t-1) + \epsilon/N} \right)$, $k_{i,t} = 0$

and $l_{i,t} = \tilde{l}_{i,t}$, and using the fact that $\beta_i \geq \mu_{i,t}$, the dual function associated with the offline problem becomes

$$\begin{aligned} &D(\mu_{i,t}, \lambda_t, l_{i,t}, k_{i,t}) = \sum_t \lambda_t D(t) \\ &+ \min_{s_i(t)} \sum_t \sum_i [f_{i,t}(s_i(t)) + (\mu_{i,t} - \mu_{i,t+1} - \lambda_t - l_{i,t}) s_i(t)] \\ &= \sum_t \lambda_t D(t) + \min_{s_i(t)} \sum_t \sum_i [f_{i,t}(s_i(t)) - f_i'(\tilde{s}_i(t)) s_i(t)] \\ &= \sum_t \lambda_t D(t) + \sum_t \sum_i \left[f_{i,t}(\tilde{s}_i(t)) - f_{i,t}'(\tilde{s}_i(t)) \tilde{s}_i(t)\right] \end{aligned} \tag{8}$$

where the last equation follows from the convexity of $f_{i,t}(\cdot)$. Putting (7) into (8) and using the weak duality and the fact that $\tilde{s}_i(t)\tilde{l}_{i,t} = 0$, we have the following

$$\begin{aligned} &\text{Offline Cost} \geq \text{value of (8)} \\ &= \sum_{t=1}^{T} \sum_{i=1}^{N} \left[f_{i,t}(\tilde{s}_i(t)) + \frac{\beta_i}{\eta} \ln\left(\frac{\tilde{s}_i(t) + \epsilon/N}{\tilde{s}_i(t-1) + \epsilon/N}\right) \tilde{s}_i(t)\right] \end{aligned}$$

Thus, the competitive ratio of Algorithm 1 becomes

$$\begin{aligned} CR &= \frac{\text{Online Cost}}{\text{Offline Cost}} \\ &\leq \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} [f_{i,t}(\tilde{s}_i(t)) + \beta_i(\tilde{s}_i(t) - \tilde{s}_i(t-1))^+]}{\sum_{t=1}^{T} \sum_{i=1}^{N} \left[f_{i,t}(\tilde{s}_i(t)) + \frac{\beta_i}{\eta} \ln\left(\frac{\tilde{s}_i(t)+\epsilon/N}{\tilde{s}_i(t-1)+\epsilon/N}\right) \tilde{s}_i(t)\right]} \\ &\leq 1 + \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \beta_i \tilde{s}_i(t)}{\sum_{t=1}^{T} \sum_{i=1}^{N} \left[f_{i,t}(\tilde{s}_i(t)) + \frac{\beta_i}{\eta} \ln\left(\frac{\tilde{s}_i(t)+\epsilon/N}{\tilde{s}_i(t-1)+\epsilon/N}\right) \tilde{s}_i(t)\right]} \\ &\leq 1 + \frac{\beta \sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{s}_i(t)}{\sum_{t=1}^{T} \sum_{i=1}^{N} \left[e_{0,i} \tilde{s}_i(t) + \frac{\beta_i}{\eta} \ln\left(\frac{\tilde{s}_i(t)+\epsilon/N}{\tilde{s}_i(t-1)+\epsilon/N}\right) \tilde{s}_i(t)\right]} \\ &\leq 1 + \frac{\beta}{e_0 + C} \end{aligned}$$

□

Theorem IV.2 shows that Algorithm 1 has a smaller competitive ratio compared to the greedy algorithm whenever $C > 0$. Although it is difficult to get the accurate value of C due to the complex correlation between ϵ and $\tilde{s}_i(t)$, we have $C = 0$ when all $s_i(t)$ are equal and $C = \beta$ when $N = T = 1$ and $\tilde{s}_i(1) = D_{max}$. Moreover, C approaches to β when the workload dispatched to each data center oscillates dramatically, a usual case in real world as shown in Section VI. Thus, the new bound in Theorem IV.2 can be close to $1 + \frac{\beta}{e_0+\beta}$, which is very helpful especially when e_0 is very small compared to β. Further, the regularization algorithm outperforms the greedy algorithm in our real-data based simulation in Section VI.

When implemented in practice, Algorithm 1 can be deployed on a central load balancer that dispatches workload to each data center based on the algorithm's result. According to the numerical results in Section VI, the regularization algorithm responds to the change of switching cost more dramatically compared to the greedy algorithm in [17]. When the switching cost is low, regularization algorithm leads to more frequent workload switching compared to the greedy algorithm. On the contrary, the workload switching is performed more conservatively as the switching cost increases.

V. Online Regularization Algorithm with Switching Cost Offset

In this section, we consider the case when $r_i(t) > 0$, that is, when there is a non-zero offset for the switching cost. We note that $r_i(t)$ can capture the saving from renewable energy access, the delay tolerance of computing workload, as well as the reduced server's wear-and-tear cost and state-toggling risk with certain protection mechanisms as discussed in Section III. Further, we assume that the operational cost function is linear in this section and let $c_i(t)$ denote the unit operational cost. Problem (1) then becomes

$$\begin{aligned} \min_{s_i(t)} &\sum_{t=1}^{T} \sum_{i=1}^{N} \left[c_i(t) s_i(t) + \beta_i (s_i(t) - s_i(t-1) - r_i(t))^+\right] \\ s.t. &\sum_{i=1}^{N} s_i(t) \geq D(t) \ \forall t \\ &s_i(t) \geq 0 \ \forall i, t \\ &s_i(0) = 0 \ \forall i \end{aligned} \tag{9}$$

The problem with a general convex operational cost and a non-zero switching offset remains open.

We first note that Algorithm 1 may perform poorly in the presence of $r_i(t)$ as shown in Section VI. Thus, we have designed a new regularization based online algorithm as shown in Algorithm 2. Compared with Algorithm 1, the main difference is that Algorithm 2 distinguishes two cases when solving the convex optimization problem (12) based on the values of K_c and K_s. When $1 \leq K_s \leq K_c$, meaning that $r_i(t)$ is relatively small, Algorithm 2 runs the same convex optimization as in Algorithm 1 since small $r_i(t)$ will not result in a big performance loss. When $K_s > K_c$ (e.g., when $r_i(t)$ is large), Algorithm 2 sets a different value of η to utilize the large switching offset for a more aggressive switching policy. Note that, Algorithm 2 needs the bound of $c_i(t)$ and $r_i(t)$. As a result, the competitive ratio of Algorithm 2 also depends on these two parameters.

Theorem V.1. *The optimal solution $\tilde{s}_i(t)$ of Algorithm 2 can achieve a competitive ratio of $\Lambda(1 + 2\ln(1 + N\frac{D_{max}}{D_{min}}))$ compared to the offline optimal solution of (9) where*

$$\Lambda = \begin{cases} K_s & \text{if } 1 \leq K_s \leq K_c \\ K_c & \text{otherwise}, \end{cases} \tag{13}$$

by setting $\epsilon = D_{min}$.

The main challenge of the proof is that the offline dual function of (9) has an extra negative term that is related to $r_i(t)$, leading to the coupling of workload dispatch decisions across multiple time slots and data centers. Therefore, we prove the competitive ratio in two cases based on the value of K_s and K_c as in Algorithm 2. We assign two different sets of dual variables in different cases. In case 1 where $1 \leq K_s \leq K_c$, the dual variables of (12) are assigned to the same values as in the proof of Theorem IV.2. In the other case where $K_s < 1$ or $K_s > K_c$, we assign different values for the dual variables. Please refer to [26] for a detailed proof.

The competitive ratio in Theorem V.1 depends on the values of β_i, $r_i(t)$ and $c_i(t)$, which we believe is necessary for all online algorithms. We plot Λ versus $r_i(t)$ in Figure 1

Algorithm 2 Online Regularization with Linear Operational Cost and Switching Cost Offset

```
Compute
    $K_c = \max\{\frac{2(1+\epsilon/D_{min})D_{max}\beta_i}{\min_{i,t}\{r_i(t)\}\min_{i,t}\{c_i(t)\}}, 1\}$    (10)
    $K_s = \frac{1}{1-\frac{\sum_i \beta_i \max_{i,t}\{r_i(t)\}}{\min_{i,t}\{c_i(t)\}D_{min}}}$    (11)
For input $\epsilon > 0$, set
    $\eta = \begin{cases} \ln(1+ND_{max}/\epsilon) & \text{if } 1 \le K_s \le K_c \\ K_c \ln(1+ND_{max}/\epsilon) & \text{o/w} \end{cases}$
Initialize $\tilde{s}_i(0) = 0$ for all $i = 1, \cdots, N$
for $t \leftarrow 1$ to $T$ do
    The ISO solves the following problem to obtain $\tilde{s}_i(t)$
    $\tilde{s}_i(t) = \text{argmin}_{s_i(t)\in P_t}\Big\{c_i(t)s_i(t) + \frac{1}{\eta}\sum_{i=1}^{N}\beta_i\Big[(\tilde{z}_i(t)+\epsilon/N)\ln\Big(\frac{\tilde{z}_i(t)+\epsilon/N}{\tilde{s}_i(t-1)+\epsilon/N}\Big) - \tilde{z}_i(t)\Big]\Big\}$    (12)
    where $P_t = \{s_i(t) | \sum_{i=1}^{T} s_i(t) \ge D(t),\ s_i(t) \ge 0\ \forall i\}$
    and
    $\tilde{z}_i(t) = \begin{cases} s_i(t) & \text{if } 1 \le K_s \le K_c \\ \max\{s_i(t) - r_i(t), \tilde{s}_i(t-1)\} & \text{o/w} \end{cases}$
end for
```

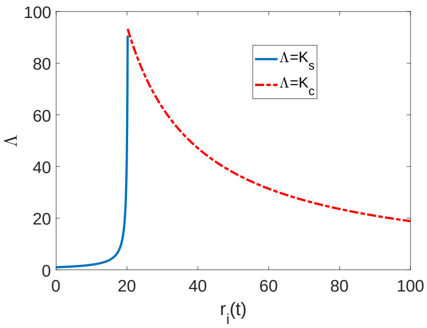


Fig. 1. Competitive Ratio with Switching Offset

with the same setting as in Section VI, where we have five data centers and use real data for unit electricity prices $c_i(t)$, as well as actual workload trace from the Google Cloud Platform as $D(t)$ and $\beta = 6$. $r_i(t)\ \forall i, t$ in Figure 1 are all equal. The blue solid line refers to the case when $\Lambda = K_s$ and the red dash line refers to the case when $\Lambda = K_c$. Figure 1 shows that Λ increases first and goes down after a certain value as $r_i(t)$ increases. When $r_i(t) = 0$, we have $\Lambda = 1$ and the competitive ratio is $1 + 2\ln(1 + N\frac{D_{max}}{D_{min}})$, similar to the bound in [7]. As $r_i(t)$ increases, Λ becomes significantly large due to (11). When $r_i(t)$ keeps increasing, we eventually have $K_s < 0$ and $K_c \approx 1$ from (10). Thus, $\Lambda \approx 1$ and the competitive ratio becomes close to the upper bound $1+2\ln(1+N\frac{D_{max}}{D_{min}})$ again. Thus, Λ is determined by K_c when $r_i(t)$ is large and is determined by K_s when $r_i(t)$ is small as illustrated by Figure 1. Note that if we directly apply Algorithm 1 to (9), the performance can be very bad for large $r_i(t)$ as shown in Section VI.

VI. Numerical Results

In this section, we evaluate the performance of our algorithms in various circumstances using real-data based simulations.

A. Simulation Setup

Our simulation is based on real-world data traces for data center locations, workload, energy prices, and renewable energy supply as discussed below.

1) The workload: We use the workload trace in May 2011 from a Google Cluster of about 12.5k machines [25] shown in Figure 2(a). We count the average number of jobs arrived at the cluster every five minutes over two days.

2) The availability of renewable energy: We use traces with 5 minutes granularity from [1], [2] for solar and wind energy in five states where Google data centers are located. Figure 2(b) shows the normalized Global Horizontal Irradiance from five solar plants and Figure 2(c) shows the normalized energy generation from five wind farms in the corresponding states.

Note that both the solar and wind energy data shown in the figures are normalized with respect to the workload. Specifically, we first find the ratio between the total renewable energy (across all the sources and the two-day time horizon) and the corresponding total workload. The value of the renewable energy supply at each time slot is then divided by this ratio. Since it is unknown how much of the renewable energy supply was available to the cluster, we further introduce a parameter ρ_o and assume that the amount of workload that each renewable energy source can serve at each time slot equals to the normalized renewable energy supply times ρ_0. Similarly, we introduce a parameter ρ_s to model the portion of the switching cost that can be covered by renewable energy supply. That is, the switching cost offset for each data center at each time slot equals to the corresponding normalized renewable energy times ρ_s.

3) The data center system: We consider a system with five data centers located in CA, IL, OR, TX and NC where Google has data centers. Each data center has access to the closest solar plant and wind turbine farm. We consider two operating cost functions in the simulation. In Figure 3, the operating cost equals to the energy price plus an extra penalty term as the following:

$$f_{i,t}(x) = (p_i + M_{i,t})x \tag{14}$$

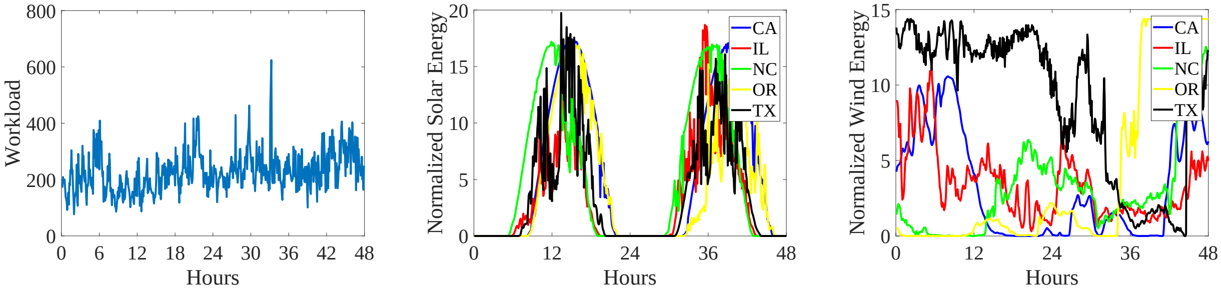
where p_i is the industrial electricity price in each of the five states in May 2011 [3] and $M_{i,t}$ is a cyclic penalty term

$$M_{i,t} = \begin{cases} 10p_i & \text{if } t \mod N \ge i \\ 0 & \text{otherwise} \end{cases}$$

In Figures 4 and 5, we consider another operational cost function consisting of the energy cost and delay cost. The energy cost is defined as follows:

$$p_i(x - \varepsilon_{i,t})^+ \tag{15}$$

where p_i is the same price as in (14) and $\varepsilon_{i,t}$ models the amount of workload that can be served by renewable energy and is set to a fixed portion (20%) of the normalized



(a) Two-day average workload from Google

(b) Solar Data

(c) Wind Data

Fig. 2. Workload and the Renewable Energy Supply

renewable energy, that is, $\rho_o = 0.2$. For the delay cost, we use a similar model as in [17]:

$$L_{i,t} = \delta_i + \frac{1000ms}{\mu_i - x}$$

where δ_i is the transmission delay between each data center and the central workload balancer (CA) resulting in delays between $10ms$ and $260ms$. $\mu_i = 0.1(ms)^{-1}$ refers to the average number of jobs processed per unit time.

We use the renewable energy in each state as the switching offset $r_i(t)$ with normalized portion $\rho_s = 0.3$ in Figure 4 and vary its value in Figure 5. For switching cost, we set $\beta = 20$ in Figure 5 and vary β in Figures 3 and 4 to show its impact on total cost in different algorithms.

B. Simulation Results

We perform several simulations to evaluate the impact of the switching cost and the switching cost offset in various circumstances. In Figure 3, we set the switching cost offset $r_i(t) = 0$ and compare the greedy algorithm and the regularization algorithm. In Figures 4 and 5, we set $r_i(t)$ to be the renewable energy supply at t and investigate its effect.

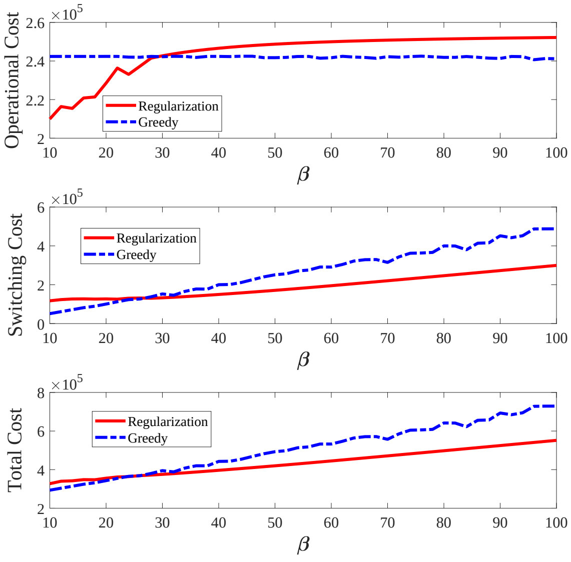


Fig. 3. Regularization Algorithm vs. Greedy Algorithm

We first compare the performance of the greedy algorithm and the regularization based algorithm when there is no switching cost offset. In Figure 3, we vary the value of the switching cost β while fixing all the other parameters. The top two subfigures show the operational and switching cost of both regularization algorithm and greedy algorithm respectively. The bottom one compares the overall performance of the two algorithms. Based on our analysis in Section IV, the greedy algorithm has a larger competitive ratio. Moreover, the real performance of the regularization algorithm is also much better than the greedy algorithm when β increases as shown in Figure 3. In addition, as β increases, the regularization algorithm reduces the amount of workload switching (difference of workload assigned to a data center between two consecutive time slots) to each data center more dramatically compared to the greedy algorithm. The workload switching for regularization algorithm is more aggressive when β is small and more conservative when β is large. We can see that the switching cost of the regularization algorithm increases much slower than the greedy algorithm, leading to a higher operational cost than the latter.

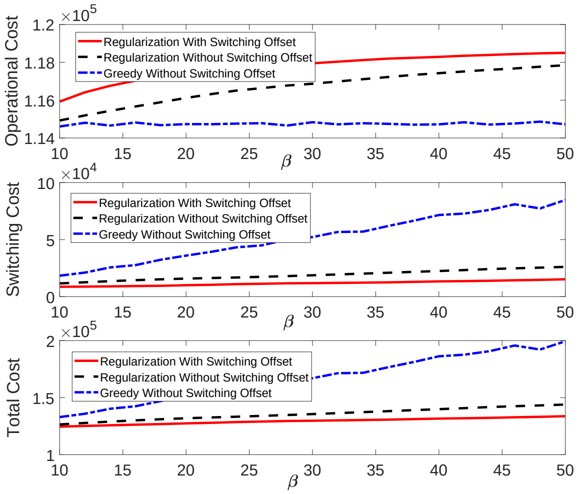


Fig. 4. Cost vs. β with Real Data

In Figure 4 and Figure 5, we consider 20% of the normalized renewable energy serving the operational cost and the switching cost offset $r_i(t)$ equals to the value that the normalized renewable energy multiplies ρ_s. In both figures,

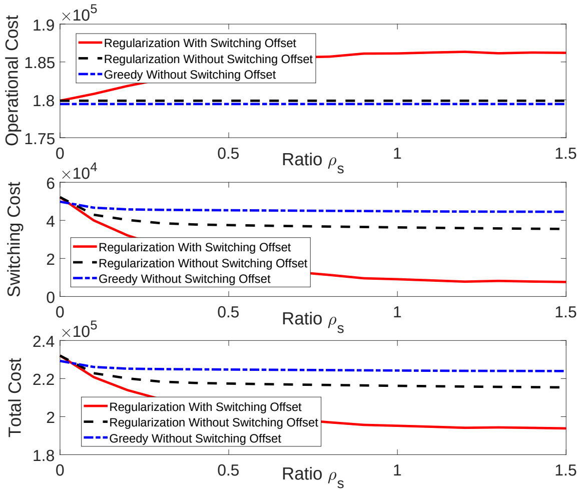


Fig. 5. Cost vs. Switching Offset with Real Data

we plot the operational cost, switching cost and total cost respectively. We compare the performance of Algorithm 2, Algorithm 1 and the greedy algorithm to investigate the impact of β and $r_i(t)$. In Figure 4, we vary the value of β. The top two sub-figures show that the two regularization algorithms (with and without considering $r_i(t)$) have higher operational cost and smaller switching cost, which is consistent with the observation in Figure 3. The bottom sub-figure shows that Algorithm 2 outperforms the other two as expected. In Figure 5, we study the impact of $r_i(t)$ by varying the value of ρ_s, which is linearly proportional to $r_i(t)$. We observe that the total cost of Algorithm 2 is much smaller than the other two algorithms and the gap increases as $r_i(t)$ becomes larger. This is expected since Algorithm 2 utilizes $r_i(t)$ to adjust the workload dispatch more aggressively to reduce the total cost. For Algorithm 1 and the greedy algorithm, the decrease in the total cost only comes from the increase of $r_i(t)$.

VII. Conclusion

In this paper, we study the right-sizing problem in a system consisting of a central workload balancer and multiple heterogeneous data centers with different operational cost and switching cost. We further introduce a switching cost offset to our model. Two online regularization algorithms are proposed for the case with and without the switching cost offset. For the case without switching cost offset, we show that our algorithm performs better than the greedy algorithm in terms of both the competitive ratio obtained and the real performance in real data based simulations. When considering the switching cost offset, our algorithms achieves a competitive ratio proportional to the logarithm of the number of data centers

References

[1] `https://www.nrel.gov/rredc/`.

[2] `https://www.nrel.gov/grid/wind-integration-data.html`.

[3] `http://www.eia.doe.gov`.

[4] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu. Energy proportional datacenter networks. In *ACM SIGARCH Computer Architecture News*, volume 38, pages 338–347, 2010.

[5] N. Bansal, A. Gupta, R. Krishnaswamy, K. Pruhs, K. Schewior, and C. Stein. A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs. In *Proc. of APPROX/RANDOM*, 2015.

[6] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12), 2007.

[7] N. Buchbinder, S. Chen, and J. S. Naor. Competitive analysis via regularization. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 436–444, 2014.

[8] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. *ACM SIGOPS operating systems review*, 35(5):103–116, 2001.

[9] S. Chen, L. Tong, and T. He. Optimal deadline scheduling with commitment. In *49th Allerton*, pages 111–118, 2011.

[10] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In *ACM SIGMETRICS*, volume 33, pages 303–314, 2005.

[11] A. K. Coskun, R. Strong, D. M. Tullsen, and T. Simunic Rosing. Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors. In *ACM SIGMETRICS*, volume 37, pages 169–180, 2009.

[12] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *ACM SIGMETRICS*, 2009.

[13] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini. Intelligent placement of datacenters for internet services. In *International Conference on Distributed Computing Systems*, pages 131–142. IEEE, 2011.

[14] T. Horvath and K. Skadron. Multi-mode energy management for multi-tier server clusters. In *International Conference on Parallel Architectures and Compilation Techniques*, pages 270–279, 2008.

[15] B. Jennings and R. Stadler. Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, 23(3):567–619, 2015.

[16] J. Koomey. Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, at the request of The New York Times*, 9, 2011.

[17] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. In *IGCC*, 2012.

[18] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM TON*, 21(5):1378–1391, 2013.

[19] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. Mobile edge computing: Survey and research outlook. *arXiv preprint arXiv:1701.01090*, 2017.

[20] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 123–134, 2009.

[21] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *Proc. of IEEE INFOCOM*, 2010.

[22] R. Stanojevic and R. Shorten. Distributed dynamic speed scaling. In *Proc. of IEEE INFOCOM*, 2010.

[23] K. Wang, M. Lin, F. Ciucu, A. Wierman, and C. Lin. Characterizing the impact of the workload on the value of dynamic resizing in data centers. In *ACM SIGMETRICS, year=2012*.

[24] R. Weingärtner, G. B. Bräscher, and C. B. Westphall. Cloud resource management: A survey on forecasting and profiling models. *Journal of Network and Computer Applications*, 47:99–106, 2015.

[25] J. Wilkes. More Google cluster data. Google research blog, Nov. 2011. Posted at `http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html`.

[26] M. Zhang, Z. Zheng, and N. Shroff. An online algorithm for power-proportional data centers with switching cost. Technical report. `http://web.cse.ohio-state.edu/~zhang.2562/cdc-main-tech-report.pdf`.

[27] X. Zhang, C. Wu, Z. Li, and F. C. Lau. Online cost minimization for operating geo-distributed cloud cdns. In *IEEE IWQoS*, 2015.

[28] S. Zhao, X. Lin, D. Aliprantis, H. N. Villegas, and M. Chen. Online multi-stage decisions for robust power-grid operations under high renewable uncertainty. In *Proc. of IEEE INFOCOM*, 2016.

[29] Z. Zheng and N. Shroff. Online welfare maximization for electric vehicle charging with electricity cost. In *Proc. of ACM e-Energy*, 2014.