

# A Sequential Game Perspective and Optimization of the Smart Grid with Distributed Data Centers

Yanzhi Wang, Xue Lin, and Massoud Pedram

Department of Electrical Engineering  
University of Southern California  
Los Angeles, USA  
{yanzhiwa, xuelin, pedram}@usc.edu

**Abstract**—The emergence of cloud computing has established a trend towards building massive, energy-hungry, and geographically distributed data centers. Due to their enormous energy consumption, data centers are expected to have major impact on the electric grid by significantly increasing the load at locations where they are built. Dynamic energy pricing policies in the recently proposed smart grid technology can incentivize the cloud computing central controller to shift the computation load towards data centers located in regions with cheaper electricity. Moreover, data centers and cloud computing also provide opportunities to help the smart grid with respect to robustness and load balancing. To gain insights into these opportunities, we consider an interaction system of the smart grid and cloud computing. We provide the sequential game formulation of the interaction system, under two different dynamic pricing scenarios: the power-dependent pricing and the time-ahead pricing. The two players in the sequential games are the smart grid controller that sets the energy price signal and the cloud computing central controller that performs resource allocation among data centers. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses, while the objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent price signal. Based on the backward induction principle, we derive the optimal or near-optimal strategies for the two players in the sequential game using convex optimization and effective heuristic search techniques. Experimental results demonstrate the effectiveness of the proposed sequential game-based optimization framework on profit maximization and load balancing.

## I. INTRODUCTION

Cloud computing has been envisioned as the next-generation computing paradigm for its advantages in on-demand service, ubiquitous network access, location independent resource pooling, and transference of risk [1]. Cloud computing shifts the computation and storage resources from the network edges to a “Cloud” from which businesses and users are able to access applications from anywhere in the world on demand [2][3]. In cloud computing, the capabilities of business applications are exposed as sophisticated services that can be accessed over a network. Cloud service providers are incentivized by the profits by charging clients for accessing these services. Clients are attracted by the opportunity for reducing or eliminating costs associated with “in-house” provision of these services.

The cloud service providers own large data centers with massive computation and storage capabilities. Service providers often end up over-provisioning their resources in

these data centers in order to meet the clients’ response time requirements or service level agreements (SLAs) [4]. Such over-provisioning may increase the cost incurred on the servers in terms of both the electrical energy cost and the carbon emission. Therefore, optimal allocation of the resources in the cloud computing system is imperative in order to reduce the cost incurred on the servers as well as the environmental impact, and has been investigated in [5][6].

The major cloud providers such as Google, Microsoft, and Amazon have built and are working on building the world’s largest data centers with enormous energy consumption. For example, Microsoft’s data center in Quincy, Washington consumes 48 megawatts which is enough to power 40,000 homes [7]. Data centers are expected to have a major impact on the electric grid by significantly increasing the load consumption at locations where they are built. Hence, integration of large-scale data centers may degrade the robustness and reliability of the traditional power grid with respect to link breakage and load demand variations.

The recently proposed smart grid technology can monitor and control the power flow in the grid to match the amount of power generation to that of the power consumption, and to minimize the overall cost of electrical power delivered to the end users [8]. Utility companies can employ *time-dependent* or *location-dependent* dynamic electricity pricing strategies incentivizing consumers to perform *demand side management* [9] by adjusting the electric loads to match the current state of the network, i.e., shifting their loads from the peak time periods to off-peak periods or from one physical location to another location. As the data centers grow in size, the cost of electricity is dominating all other cost aspects in cloud computing. Central controller of the cloud computing system should develop resource management algorithms among data centers that take into account the changes in the electricity price during the day at different regions by dynamically shifting the computation load towards data centers which are located in regions with cheaper electricity. Developing such resource management algorithms and location-dependent dynamic electricity pricing strategies are important in order for mitigating the negative impacts on the power grid from integrating large-scale data centers. With appropriately designed dynamic pricing policies, it is even possible that cloud computing and data centers could actually help the smart grid design in terms of *load balancing* and robustness thanks to the flexibility in *service request dispatching* to various data centers [10].

In this paper, we consider a smart grid system consisting of multiple power buses. We consider a set of distributed data centers in this infrastructure. Each data center, which consists of potentially heterogeneous servers in terms of request processing ability, is connected to one bus in the power grid to obtain the electricity needed for its operation. Service requests from a common *request pool* are free to be dispatched to any server in the cloud computing system. The total profit in the cloud computing system is the total price gained from serving the service requests, which depends on the average request response time as defined in the *utility function*, subtracted by the energy cost of the active datacenters and servers.

We consider two different location-dependent dynamic pricing scenarios: *power-dependent pricing* [11] and *time-ahead pricing* [12]. In the first case, the smart grid controller announces the price signal based on the power consumptions of load devices connected to different power buses, while in the second case, the smart grid controller announces the price signal first and the cloud computing system and other users perform demand side management as response. We consider the interaction system of smart grid and cloud computing and provide the sequential game formulations with two players, i.e., the smart grid controller and the cloud computing central controller, under these two scenarios. In the first scenario, the cloud computing controller is the first player and the smart grid controller is the second. In the second scenario they are the opposite. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses. The objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent price signal. We derive the optimal or near-optimal strategies for both players based on backward induction [17], using convex optimization [16] and effective heuristic search approaches. Experimental results demonstrate the effectiveness of the proposed sequential game-based optimization framework on profit maximization and load balancing.

The rest of this paper is organized as follows. The smart grid model and resource allocation model in cloud computing are introduced in Section II. The sequential game formulation and optimization under the two dynamic pricing scenarios are provided in Section III and Section IV, respectively. Experimental results and conclusion are presented in Section V and Section VI, respectively.

## II. SYSTEM MODEL

### A. The Smart Grid

Consider a smart power grid and let  $\mathcal{N}$  with size  $N$  denote the set of all *power buses*, indexed by  $i$ . The power buses are interconnected through *branches* forming the grid topology. Each bus  $i \in \mathcal{N}$  is connected to various load devices. In our system model, some loads of the power grid may include large data centers which support cloud computing. There are  $M$  distributed data centers in this infrastructure, indexed by  $j$ . Each data center is connected to one power bus in the power grid to obtain the electricity required for its operation. We use

$bus(j)$  to denote the index of the power bus that the  $j$ -th data center is connected to.

The total load power consumption at the  $i$ -th power bus, denoted by  $P_{bus,i}^{Load}$ , is calculated via:

$$P_{bus,i}^{Load} = P_{bus,i}^{DC} + P_{bus,i}^{Back}, \quad (1)$$

where  $P_{bus,i}^{DC}$  denotes the total power consumption of the data centers (if any) connected to bus  $i$ ; and the term  $P_{bus,i}^{Back}$  denotes the power consumption of any load other than data centers at bus  $i$ . Let  $P_{DC,j}$  denote the power consumption of the  $j$ -th data center. Then  $P_{bus,i}^{DC}$  is calculated by

$$P_{bus,i}^{DC} = \sum_{bus(j)=i} P_{DC,j}. \quad (2)$$

If there is no datacenter connected to bus  $i$ , we have  $P_{bus,i}^{DC} = 0$ . We further define power consumption vectors  $\mathbf{P}_{bus}^{Load} = \{P_{bus,1}^{Load}, P_{bus,2}^{Load}, \dots, P_{bus,N}^{Load}\}$ ,  $\mathbf{P}_{bus}^{DC} = \{P_{bus,1}^{DC}, P_{bus,2}^{DC}, \dots, P_{bus,N}^{DC}\}$ , and  $\mathbf{P}_{bus}^{Back} = \{P_{bus,1}^{Back}, P_{bus,2}^{Back}, \dots, P_{bus,N}^{Back}\}$ .

We consider location-dependent dynamic pricing in this paper, i.e., the smart grid controller announces different prices for different power buses. We consider two different pricing scenarios: *power dependent pricing* and *time-ahead pricing*. In the power-dependent pricing scenario, the smart grid controller announces the price signal vector, denoted by  $\mathbf{price}(\mathbf{P}_{bus}^{Load})$ , based on the power consumption vector  $\mathbf{P}_{bus}^{Load}$ . In order to perform load balancing by incentivizing the cloud computing controller to shift the loads among data centers, we assume that the unit energy price for the  $i$ -th ( $1 \leq i \leq N$ ) bus is proportional to  $P_{bus,i}^{Load}$ , i.e.,

$$price_i(\mathbf{P}_{bus}^{Load}) = C \cdot P_{bus,i}^{Load}, \quad (3)$$

where  $C$  is a constant. This pricing scheme is similar to [11]. Moreover, the average price is regulated by the government or other administration agency, i.e.,

$$\frac{\sum_{1 \leq i \leq N} price_i(\mathbf{P}_{bus}^{Load})}{N} \leq price_{avg,max}. \quad (4)$$

In this pricing scenario, the other load devices than the data centers cannot perform demand side management (i.e., the  $P_{bus,i}^{Back}$  values are fixed) due to the lack of *a priori* knowledge of the price signal.

In the time-ahead pricing scenario, the smart grid controller announces the price signal first and the cloud computing system and other users (load devices) perform demand side management as response. In order for better performing load balancing, the smart grid controller employs a *dual price* scheme similar to [14], i.e., it utilizes two potentially different unit energy prices, i.e.,  $price_i$  and  $price'_i$ , for the data centers and other load devices connected to the  $i$ -th ( $1 \leq i \leq N$ ) power bus, respectively. Intuitively, the peak of data center power consumption will cancel the trough of other load devices' power consumption at each power bus under this dual-pricing policy. We also add regulations so that the average prices for the data centers and for other load devices should not exceed  $price_{avg,max}$  and  $price'_{avg,max}$ , respectively.

The cloud computing controller and other load devices will perform demand side management accordingly. The cloud computing controller determines the data center power consumption at each  $i$ -th ( $1 \leq i \leq N$ ) power bus, denoted by  $P_{bus,i}^{DC}(\mathbf{price})$ , based on a joint consideration of the price vector  $\mathbf{price} = \{price_1, price_2, \dots, price_N\}$  for all the power buses, as shall be discussed later. On the other hand, the other load devices will use distributed storage systems [12] or other load shaping techniques [13] to reduce the power consumption when the unit energy price is high. We assume a linearly decreasing relationship between  $P_{bus,i}^{Back}$  and  $price'_i$ , i.e.,

$$P_{bus,i}^{Back}(price'_i) = P_{bus,i}^{Back}(0) - \alpha_i \cdot price'_i. \quad (5)$$

### B. Resource Allocation in the Cloud Computing System

Figure 1 shows the structure of the target resource allocation system in cloud computing with a service request pool,  $M$  distributed data centers as well as a central resource management node. Each  $j$ -th data center consists of  $K_j$  potentially heterogeneous servers. We use  $k$  as the index of servers in a data center.

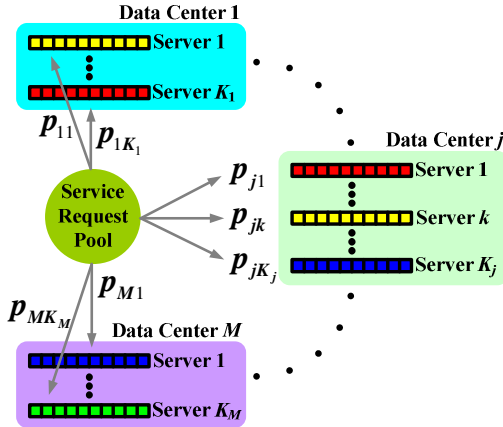


Figure 1. Architecture of the resource allocation problem in the cloud computing system.

The service request pool contains service requests generated from all the clients. A service request can be dispatched to any server in the cloud computing system. The request dispatcher assigns a request to the  $k$ -th server in the  $j$ -th data center with probability  $p_{jk}$ . These probability values are the optimization variables in the resource allocation optimization framework.

In order to find the analytical form of the average response time, service requests are assumed to follow a Poisson process with an average generating rate of  $\lambda$  (predicted based on the behavior of the clients.) According to the properties of the Poisson distribution, service requests that are dispatched to the  $k$ -th server in the  $j$ -th data center follow a Poisson process with an average rate of  $p_{jk} \cdot \lambda$ , which is the average service request arrival rate of that server.

Each  $k$ -th server in the  $j$ -th data center allocates a portion of its total resources, denoted by  $\phi_{jk}$  ( $0 \leq \phi_{jk} \leq 1$ ), for servicing the requests. By using the well-known formula in M/M/1 queues [15], the average response time of service requests dispatched to that server is calculated as

$$R_{jk}(p_{jk}, \phi_{jk}) = \begin{cases} \frac{1}{\phi_{jk} \cdot \mu_{jk} - p_{jk} \cdot \lambda} & \text{if } p_{jk} > 0, \\ 0 & \text{if } p_{jk} = 0, \end{cases} \quad (6)$$

where  $\mu_{jk}$  denotes the average service request processing speed when all the resources in the server are allocated for request processing.

Power consumption in each server consists of a *dynamic power consumption* part when the server is active (i.e., when it is processing service requests) and a *static power consumption* part. The average dynamic power consumption in each  $k$ -th server in the  $j$ -th data center is proportional to the portion of time that the server is active, given by  $(p_{jk} \cdot \lambda) / (\phi_{jk} \cdot \mu_{jk})$ , as well as the portion  $\phi_{jk}$  of the resources that have been allocated for request processing:

$$P_{Serv,jk}^{dyn}(p_{jk}) = \frac{p_{jk} \cdot \lambda}{\phi_{jk} \cdot \mu_{jk}} \cdot \phi_{jk} \cdot P_{Serv,jk}^{dyn,max} \\ = \frac{p_{jk} \cdot \lambda}{\mu_{jk}} \cdot P_{Serv,jk}^{dyn,max}, \quad (7)$$

where  $P_{Serv,jk}^{dyn,max}$  is the dynamic power consumption when the server is active and all resources have been allocated for service request processing. On the other hand, the (average) static power consumption in each  $k$ -th server in the  $j$ -th data center is the sum of a constant term  $\varepsilon_{jk}$  and another term proportional to the portion  $\phi_{jk}$  of allocated resources for request processing:

$$P_{Serv,jk}^{sta}(\phi_{jk}) = \varepsilon_{jk} + \phi_{jk} \cdot (P_{Serv,jk}^{sta,max} - \varepsilon_{jk}). \quad (8)$$

The power consumption of each  $j$ -th data center is the sum of the total power consumption of all its servers, i.e.,

$$P_{DC,j} = \sum_{1 \leq k \leq K_j} (P_{Serv,jk}^{dyn}(p_{jk}) + P_{Serv,jk}^{sta}(\phi_{jk})). \quad (9)$$

Let  $U(R) = \beta - \gamma \cdot R$  denote the utility function of the cloud computing system with the average service request response time equal to  $R$ . Then the total profit of the cloud computing system is calculated by<sup>1</sup>:

$$\lambda \cdot \left( \beta - \gamma \cdot \sum_{j=1}^M \sum_{k=1}^{K_j} \frac{p_{jk}}{\phi_{jk} \cdot \mu_{jk} - p_{jk} \cdot \lambda} \right) - \\ \sum_{i=1}^N price_i \sum_{bus(j)=i} \sum_{k=1}^{K_j} (P_{Serv,jk}^{dyn}(p_{jk}) + P_{Serv,jk}^{sta}(\phi_{jk})). \quad (10)$$

### III. OPTIMIZATION UNDER POWER-DEPENDENT PRICING

We consider the interaction system of smart grid and cloud computing in the power-dependent pricing scenario, and provide the sequential game formulation consisting of two players. The cloud computing central controller is the first player and the smart grid controller is the second player.

We know that the smart grid controller always sets the price  $price_i$  proportional to  $P_{bus,i}^{Load}$  under this pricing scenario

<sup>1</sup> Note that Eqn. (10) is valid when  $p_{jk} = 0$ .

as shown in (3), (4), which fact is known to the cloud computing controller. The objective of the cloud computing central controller is to maximize its own profit with an anticipation of the price signal vector from the smart grid. We name this profit maximization problem the *Resource Allocation with Anticipation of the Price signal* (RAAP) problem. The control variables of the cloud computing controller are  $p_{jk}$ 's and  $\phi_{jk}$ 's. The other parameters are either constants or functions of these control variables.

Based on the backward induction principle in sequential games [17], the cloud computing central controller maximizes the following objective function in the RAAP problem:

$$\lambda \cdot \left( \beta - \gamma \cdot \sum_{j=1}^M \sum_{k=1}^{K_j} \frac{p_{jk}}{\phi_{jk} \cdot \mu_{jk} - p_{jk} \cdot \lambda} \right) - \sum_{i=1}^N C \cdot \left( \sum_{bus(j)=i} P_{DC,j} + P_{bus,i}^{Back} \right) \cdot \left( \sum_{bus(j)=i} P_{DC,j} \right). \quad (11)$$

where  $P_{DC,j}$  depends on the control variables  $p_{jk}$ 's and  $\phi_{jk}$ 's and is given by (9). The  $P_{bus,i}^{Back}$  values are fixed in the power-dependent pricing scenario. Moreover, the pricing parameter  $C$  satisfies the following inequality constraint (derived from (4)):

$$\frac{C \cdot \sum_{1 \leq i \leq N} (\sum_{bus(j)=i} P_{DC,j} + P_{bus,i}^{Back})}{N} \leq price_{avg,max}. \quad (12)$$

The constraints of the RAAP problem are:

$$0 \leq p_{jk} \leq 1, \quad \text{for } \forall j, k, \quad (13)$$

$$0 \leq \phi_{jk} \leq 1, \quad \text{for } \forall j, k, \quad (14)$$

$$\sum_{j=1}^M \sum_{k=1}^{K_j} p_{jk} = 1, \quad (15)$$

$$p_{jk} \cdot \lambda < \phi_{jk} \cdot \mu_{jk}, \quad \text{for } \forall j, k, \quad (16)$$

where constraints (13) and (14) specify the domains of the optimization variables. Constraint (15) ensures that all service requests are serviced. Constraint (16) shows the upper limit on the average service request arrival rate to a server, i.e., it should be smaller than the average service request processing rate of that server.

The overall RAAP problem is a nonlinear programming problem and cannot be solved using conventional convex optimization methods since the objective function (11) is neither convex nor concave. We propose an iterative near-optimal solution of this optimization problem as shown in Algorithm 1. At each iteration, Algorithm 1 has an *optimal resource allocation* phase, an *optimal request dispatching* phase, and a *price parameter updating* phase as follows:

**The Optimal Resource Allocation Phase:** In this phase, the controller finds the optimal  $\phi_{jk}$ 's in order to maximize (11) when the  $p_{jk}$  values and the parameter  $C$  are given. The constraints are (14) and (16). This problem is a convex optimization problem since the objective function (11) is a concave function of  $\phi_{jk}$ 's when the  $p_{jk}$  values and parameter  $C$  are given, and constraints (14), (16) are linear inequality constraints. It can be solved optimally with polynomial time

complexity using standard convex optimization techniques. Note that when  $p_{jk} = 0$ , it is possible that the optimal  $\phi_{jk}$  value is infinitesimal. In order to find the valid  $\phi_{jk}$  values, we add the following constraint when solving this optimization problem:

$$\phi_{jk} \geq \delta, \quad \text{for } \forall j, k, \quad (17)$$

where  $\delta \ll 1$  is a small positive value.

**The Optimal Request Dispatching Phase:** In this phase, the controller finds the optimal  $p_{jk}$ 's to maximize (11) when the  $\phi_{jk}$  values and the parameter  $C$  are given. The constraints are (13), (15), and (16). This problem is also a convex optimization problem since the objective function (11) is a concave function of  $p_{jk}$ 's when the  $\phi_{jk}$  values and  $C$  are given, and therefore, it could be solved optimally with polynomial time complexity using standard techniques.

**The Price Parameter Updating Phase:** The price parameter  $C$  depends on the power consumption  $P_{bus,i}^{Load}$  at each power bus, and therefore, depends on the  $\phi_{jk}$  and  $p_{jk}$  values. At the end of each iteration (i.e., after the  $\phi_{jk}$  and  $p_{jk}$  values have been updated), we update the parameter  $C$  as the maximum  $C$  value such that (12) is satisfied. Updating the parameter  $C$  is equivalent to updating the anticipated price signal vector from the smart grid.

---

#### Algorithm 1: Near-Optimal Solution of the RAAP Problem.

---

**Initialize** the  $p_{jk}$  values and  $C$ .

**Do** the following procedure iteratively:

Optimal resource allocation: Find the optimal  $\phi_{jk}$  values that maximize (11) based on the derived  $p_{jk}$ 's and  $C$ .

Optimal request dispatching: Find the optimal  $p_{jk}$  values that maximize (11) based on the derived  $\phi_{jk}$ 's and  $C$ .

Price parameter updating: Find the maximum  $C$  value such that (12) is satisfied, and update the parameter  $C$  accordingly.

**Until** the solution converges.

---

## IV. OPTIMIZATION UNDER TIME-AHEAD PRICING

In the time-ahead pricing scenario, we consider the interaction system of smart grid and cloud computing and provide the sequential game formulation consisting of two players. Different from the power-dependent pricing scenario discussed in Section III, the smart grid controller is the first player and the cloud computing central controller is the second player in this pricing scenario.

The objective of the smart grid controller (the first player) is to achieve an optimal balance between maximizing its own profit and performing load balancing among power buses, with an anticipation of the demand side managements performed by various load devices including data centers. Based on the backward induction principle [17], the smart grid controller aims to find the optimal dual price vectors  $\mathbf{price} = \{price_1, price_2, \dots, price_N\}$  and  $\mathbf{price}' = \{price'_1, price'_2,$

...,  $price'_N$  in order to maximize the following objective function:

$$w_1 \cdot \sum_{i=1}^N \left( price_i \cdot P_{bus,i}^{DC}(\mathbf{price}) + price'_i \cdot P_{bus,i}^{Back}(price'_i) \right) - w_2 \cdot Var_{1 \leq i \leq N} \left( P_{bus,i}^{DC}(\mathbf{price}) + P_{bus,i}^{Back}(price'_i) \right), \quad (18)$$

where  $P_{bus,i}^{DC}(\mathbf{price})$  and  $P_{bus,i}^{Back}(price'_i)$  ( $1 \leq i \leq N$ ) are the power consumption values after the load devices (including data centers) have performed demand side managements based on the dual price vectors.  $Var_{1 \leq i \leq N} \left( P_{bus,i}^{DC}(\mathbf{price}) + P_{bus,i}^{Back}(price'_i) \right)$  denotes the variance of the  $P_{bus,i}^{Load} = P_{bus,i}^{DC}(\mathbf{price}) + P_{bus,i}^{Back}(price'_i)$  values for  $1 \leq i \leq N$ . A smaller variance value, which implies better load balancing, is preferred for robustness concerns.  $w_1$  and  $w_2$  are relative weights greater than or equal to zero. The constraints of the optimization problem are that the average unit energy prices for data centers and for other load devices (i.e., the average values in  $\mathbf{price}$  and  $\mathbf{price}'$ , respectively) should not exceed  $price_{avg,max}$  and  $price'_{avg,max}$ , respectively.

We name this optimization problem performed in the smart grid controller the *Optimal Pricing with Anticipation of Demand side managements* (OPAD) problem. We introduce a near-optimal solution of the OPAD problem in the following.

#### Near-Optimal Solution of the OPAD Problem:

Suppose that the price vector  $\mathbf{price}$  is announced by the smart grid controller, then the objective of the cloud computing controller is to maximize its total profit given by (10). The optimization variables are  $p_{jk}$ 's and  $\phi_{jk}$ 's. This profit maximization problem in cloud computing is a simplified version of the RAAP problem defined in Section III since  $\mathbf{price}$  is given in prior. However, it is still not a convex optimization problem. We propose an iterative near-optimal solution similar to Algorithm 1. Each iteration in the near-optimal solution consists of an optimal resource allocation phase that finds the optimal  $\phi_{jk}$ 's with given  $p_{jk}$  values, and an optimal request dispatching phase that finds the optimal  $p_{jk}$ 's with given  $\phi_{jk}$  values. We solve a convex optimization problem with polynomial time complexity in each phase. Details of the algorithm are omitted due to space limitation. We calculate  $P_{bus,i}^{DC}(\mathbf{price})$  at each  $i$ -th ( $1 \leq i \leq N$ ) power bus using (2), (9), based on the  $p_{jk}$  and  $\phi_{jk}$  values obtained from the above profit maximization problem. On the other hand, when the other price vector  $\mathbf{price}'$  is announced by the smart grid controller, the power consumption of the other load devices than the data centers at each  $i$ -th ( $1 \leq i \leq N$ ) power bus, i.e.,  $P_{bus,i}^{Back}(price'_i)$ , is calculated using (5).

Since the OPAD problem is integrated with a profit maximization problem in cloud computing, it is not possible to derive the analytical form of  $P_{bus,i}^{DC}(\mathbf{price})$  as a function of the price vector  $\mathbf{price}$ . Therefore, the OPAD problem is a hard problem to be solved optimally in polynomial time. We propose to use the simulated annealing method to find a near-optimal solution of the OPAD problem with details provided in Algorithm 2.

---

#### Algorithm 2: Near-Optimal Solution of the OPAD Problem.

---

**Initialize** the temperature  $T$ .

**Initialize**  $Obj_{max}$  to be a large negative number.

**Do** the following procedure:

Randomly change the price vectors  $\mathbf{price}$  and  $\mathbf{price}'$  satisfying the average price constraints.

**Initialize** the  $p_{jk}$  values.

**Do** the following procedure iteratively:

Optimal resource allocation: Find the optimal  $\phi_{jk}$ 's that maximize (10) based on the derived  $p_{jk}$  values and  $\mathbf{price}$ .

Optimal request dispatching: Find the optimal  $p_{jk}$ 's that maximize (10) based on the derived  $\phi_{jk}$  values and  $\mathbf{price}$ .

**Until** the solution converges.

Calculate  $P_{bus,i}^{DC}(\mathbf{price})$  for  $1 \leq i \leq N$  using (2), (9), based on the derived  $\phi_{jk}$  and  $p_{jk}$  values.

Calculate  $P_{bus,i}^{Back}(price'_i)$  for  $1 \leq i \leq N$  using (5).

$Obj \leftarrow$  the value of the objective function (18) based on the calculated  $P_{bus,i}^{DC}(\mathbf{price})$  and  $P_{bus,i}^{Back}(price'_i)$  values.

**If**  $Obj \geq Obj_{max}$ : Accept the change of  $\mathbf{price}$  and  $\mathbf{price}'$ .

**Else**: Accept the change with probability  $e^{(Obj-Obj_{max})/T}$ .

$Obj_{max} \leftarrow Obj$  if the change has been accepted.

Decrease the temperature  $T$ .

**Until** the temperature  $T$  has decreased to a certain value, i.e., the algorithm has cooled down.

---

## V. EXPERIMENTAL RESULTS

In this section, we implement the interaction system of smart grid and cloud computing and demonstrate the effectiveness of the proposed sequential game-based optimization framework.

We consider a smart power grid consisting of 12 power buses. We consider 4 data centers in the interaction system, comprised of 5 servers, 8 servers, 12 servers, and 15 servers, respectively. Each data center is connected to a power bus. We use normalized amounts of most of the parameters in the system instead of their real values. In the cloud computing system, the average service request generating rate is assumed to be 30. The maximum average service request processing rate  $\mu_{jk}$  in each server (i.e., when all its resources are allocated for request processing) is a uniformly distributed random variable between 1 and 2. The maximum dynamic power consumption  $P_{serv,jk}^{dyn,max}$  of each server is uniformly distributed between 1.5 and 3. The maximum static power consumption  $P_{serv,jk}^{sta,max}$  of each server is a uniformly distributed random variable between 0.5 and 1. For the utility function in the cloud computing system, parameter  $\beta$  is set to 9 and  $\gamma$  is 1. For the other load devices than the data centers, the parameter  $P_{bus,i}^{Back}(0)$  is a uniformly distributed random variable between 10 and 20 if a data center is connected to bus  $i$ , and is uniformly distributed between 20 and 40 if no data center is connected. The  $\alpha_i$  parameters are set to be 6.

In the first experiment, we consider the interaction system under the power-dependent pricing scenario. We compare the profit maximization capability of the cloud computing system using the proposed sequential game-based optimization method and baseline algorithm. The baseline system distributes the service requests with equal probability to each server in the cloud computing system. Figure 2 illustrates the normalized total profit versus the (maximum) average unit energy price  $price_{avg,max}$ . We can observe that the proposed method consistently outperforms the baseline algorithm. When  $price_{avg,max} = 1.2$ , the total profit of the cloud computing system obtained by the proposed optimization method is 46.4% higher than the baseline algorithm. When the average unit energy price is 1.6 or more, the total profit in the baseline system drops below zero, and is thereby not even comparable with the proposed near-optimal method.

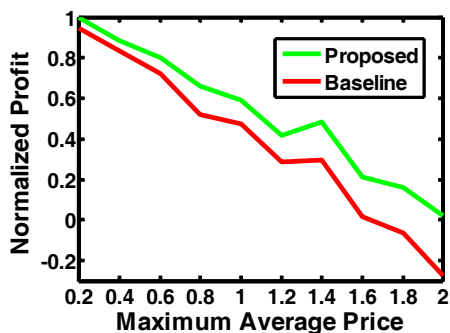


Figure 2. The normalized total profit versus the maximum average unit energy price of the proposed near-optimal method and baseline algorithm.

In the second experiment, we consider the interaction system under the time-ahead pricing scenario. We compare the capability in profit maximization and load balancing of the smart grid system using the proposed sequential game-based optimization method and baseline algorithm. The baseline algorithm sets uniform price for data centers and other load devices over all power buses. Figure 3 illustrates the tradeoff curve (obtained by adjusting parameters  $w_1$  and  $w_2$ ) between higher profit and lower variation in load power consumption at different power buses. We can observe that simultaneous enhancement in total profit and reduction in variance of load power consumption is achieved using the proposed near-optimal method. The proposed sequential game-based optimization method is extremely powerful in performing load balancing, i.e., it reduces the variance in load power consumption at different power buses by a factor up to 50X.

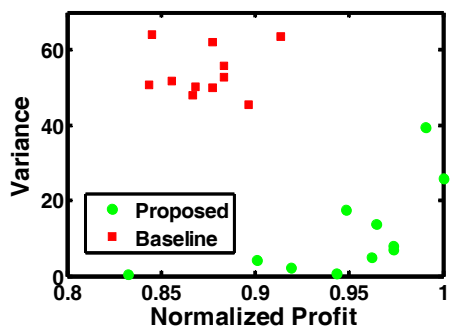


Figure 3. The tradeoff between total profit of the smart grid system and variance in load power consumption at different power buses.

## VI. CONCLUSION

In this paper, we consider an interaction system of smart grid and cloud computing. The smart grid could employ dynamic energy pricing policies to incentivize the cloud computing central controller to shift the computation load towards data centers located in regions with cheaper electricity. Data centers and cloud computing also provide opportunities to help the smart grid with respect to robustness and load balancing. We provide the sequential game formulation of the interaction system under two different dynamic pricing scenarios: the power-dependent pricing and the time-ahead pricing. The two players in the sequential games are the smart grid controller that sets the energy price signal and the cloud computing central controller that performs resource allocation among data centers. The objective of the smart grid controller is to maximize its own profit and perform load balancing among power buses, while the objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent price signal. Based on the backward induction principle, we derive the optimal or near-optimal strategies of the two players in the sequential game using convex optimization and effective heuristic search techniques.

## REFERENCES

- [1] B. Hayes, "Cloud Computing," *Communications of the ACM*, 2008.
- [2] R. Buyya, "Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility," *Proc. of CCGrid*, 2009.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Pabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, 2010.
- [4] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, 2007.
- [5] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," *IEEE Cloud*, 2011.
- [6] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing," *Proc. of CCGrid*, 2012.
- [7] R. H. Katz, "Tech Titans Building Boom," *IEEE Spectrum*, 2009.
- [8] S. Kishore, L. V. Snyder, "Control mechanisms for residential electricity demand in SmartGrids," *IEEE SmartGridComm*, 2010.
- [9] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the Smart Grid," *Proc. of IEEE SmartGridComm*, 2010.
- [10] A. H. Mohsenian-Rad and A. Leon-Garcia, "Coordination of cloud computing and smart power grids," *IEEE SmartGridComm*, 2010.
- [11] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings, "Agent-based micro-storage management for the smart grid," *Proc. of AAMAS*, 2010.
- [12] Y. Wang, S. Yue, L. Kerofsky, S. Deshpande, and M. Pedram, "A hierarchical control algorithm for managing electrical energy storage systems in homes equipped with PV power generation," *Proc. of IEEE Green Technologies Conference (GTC)*, 2012.
- [13] T. Cui et al., "Concurrent optimization of consumer's electrical energy bill and producer's power generation cost under a dynamic pricing model," *Proc. of IEEE PES ISGT*, 2012.
- [14] S. Yue et al., "Dual-pricing policy for controller-side strategies in demand side management", *Proc. of IEEE SmartGridComm*, 2011.
- [15] L. Kleinrock, *Queueing Systems, Volume I: Theory*, Wiley, 1975.
- [16] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [17] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, Morgan & Claypool Publishers, 2008.