A Stackelberg Game-Based Optimization Framework of the Smart Grid with Distributed PV Power Generations and Data Centers

Yanzhi Wang, *Student Member, IEEE*, Xue Lin, *Student Member, IEEE*, and Massoud Pedram, *Fellow, IEEE*

***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 a major impact on the electric power grid by significantly increasing the load at locations where they are built. Dynamic energy pricing policies in the recently proposed smart power grid technology can incentivize the cloud controller to shift the computation load towards data centers in regions with cheaper electricity or with excessive electricity generated by renewable energy sources, e.g., photovoltaic (PV) and wind power. On the other hand, distributed data centers in the cloud also provide opportunities to help the power grid with distributed renewable energy sources to improve robustness and load balancing. To shed some light into these opportunities, this paper considers an interaction system of the smart power grid with distributed PV power generation and the cloud computing system, jointly accounting for the service request dispatch and routing problem in the cloud with the power flow analysis in power grid. The Stackelberg (sequential) game formulation is provided for the interaction system, under two different dynamic pricing scenarios: real-time power-dependent pricing and time-ahead pricing. The two players in the Stackelberg games are the power grid controller that sets the pricing signal and the cloud controller that performs resource allocation among data centers. The objective of the power grid controller is to maximize its own profit and perform load balancing among power buses, i.e., minimizing the power flow from one power bus to the others, whereas the objective of the cloud computing controller is to maximize its own profit with respect to the location-dependent pricing signal. Based on the backward induction method, this paper derives the near-optimal or sub-optimal strategies of the two players in Stackelberg game using convex optimization and simulated annealing techniques.**

I. Introduction

Cloud computing has been envisioned as the next-generation computing paradigm for its advantages in on-demand service, location independent resource pooling, ubiquitous network access, and transference of risk [1][2]. Cloud computing transforms 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 [3][4][5]. 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 of 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 underlying infrastructure of cloud computing consists of large data centers and clusters of servers that are monitored and maintained by the cloud service providers [6]. Service providers often end up over-provisioning their resources in these servers in order to meet the clients' response time requirements or service level agreements (SLAs) [7]. Such over-provisioning may increase the cost incurred on the servers in terms of both the electrical energy cost and the carbon footprint. Hence, optimal provisioning or allocation of the resources in the cloud computing system, or in the broader area of distributed computing systems, is imperative in order to reduce the energy cost incurred on the servers as well as the environmental impact while satisfying the clients' SLAs. This topic has been extensively investigated in references [8]-[17].

The major cloud providers such as Microsoft, Google, and Amazon have built and are working on building the world's largest data centers with enormous energy consumption. A typical data center is comprised of hundreds of thousands of computer servers, cooling equipments, and substation transformers. For example, data center of Microsoft in Quincy, Washington consumes 48 megawatts that is enough to power 40,000 homes [6][28]. It is estimated that the total electricity cost of servers and data centers in the United States is $7.4 Billion annually [18], and is dominating all other cost aspects in cloud computing. 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.

The current smart power grid technology is undergoing a transformation from a centralized, producer-controlled network to one that is less centralized and more consumer-interactive, thereby minimizing the overall cost of electrical power delivered to the end users [19], [20]. Utility companies can employ *time-dependent* or *location-dependent* dynamic pricing strategies incentivizing the consumers to perform *demand side management* (DSM) [21], [22] by shifting their loads from the peak time periods to off-peak periods or from one physical location to another location. When the power grid is integrated with distributed renewable power generations such as photovoltaic (PV) or wind power, the

This research is sponsored in part by a grant from the U.S. National Science Foundation.

Y. Wang, X. Lin, and M. Pedram are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA (e-mail: yanzhiwa@usc.edu, xuelin@usc.edu, pedram@usc.edu).

A preliminary version of this paper was presented at the 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), Washington D.C., Feb. 2013 [1].

dynamic pricing policy is extremely useful for balancing the power supply and demand at different locations so as to perform proper frequency regulation [23]-[25]. In this way, such distributed renewable power generation facilities can be effectively integrated into the smart power grid despite their intermittent nature.

The cost of electricity is dominating all other types of costs in the cloud computing system. The central controller of the cloud should develop resource management algorithms among data centers that take advantage of price diversity in the deregulated electricity markets to develop algorithms that distribute the workload among data centers in multiple locations to minimize the total cost of electricity of the data centers [26], [27]. The key idea is to constantly monitor the electricity prices at different regions and shift/route the computation workloads towards data centers that are located in regions with cheaper electricity or with excessive generated electrical energy from the renewable energy sources. In this way, the cloud controller can control the impact of data centers' energy consumption on the power grid at different locations. With appropriately designed dynamic pricing policies, it is even possible that cloud computing system could actually help the power system design with distributed renewable power generations in terms of *load balancing* and robustness thanks to the flexibility in *service request dispatching* to various data centers [1][28].

In this work, we jointly consider the service request dispatch and routing problem in the cloud with the *power flow analysis* [33] in smart power grid. We consider a power grid comprised of multiple *power buses* with distributed PV power generation facilities connected to certain power buses. The buses are interconnected through *branches* forming the grid topology. Each data center, which contains potentially heterogeneous servers in terms of request processing capability and power consumption, is connected to one bus in the power grid to obtain the electricity required for its operation. Service requests from distributed clients (of the cloud computing system) are free to be dispatched to any server in the cloud. The total profit in the cloud is the total revenue gained from serving the service requests, which depends on the average request response time as defined in the *utility function* of each specific client, subtracted by the total energy cost of the active datacenters and servers.

We consider two different location-dependent dynamic pricing scenarios: real-time *power-dependent pricing* [31] and *time-ahead pricing* [29]. In the first case, the pricing signal announced by the power grid controller is dependent on the power consumption values of load devices and power generation values of distributed PV systems connected to different power buses. In the second case, the power grid controller announces the pricing signal first and the cloud computing system and other users perform demand side management in response. We consider the interaction system of smart power grid with distributed PV power generation and the cloud computing system. We provide the Stackelberg (sequential) game formulations with two players, i.e., the power grid controller and the cloud controller, under these two scenarios. In the first scenario, the cloud controller is the leading player and the power grid controller is the following player. In the second scenario they are the opposite. The objective of the power grid controller is to maximize its own profit and perform load balancing, i.e., minimizing the amount of power flowing from one power bus to another based on the power flow analysis results. The objective of the cloud controller is to maximize its own profit with respect to the location-dependent pricing signal. We derive the near-optimal or sub-optimal strategies for both players based on the backward induction method [38], using convex optimization [39] and the simulated annealing approach [41].

Experimental results on IEEE 24-bus Reliability Test System [32] demonstrate the effectiveness of the proposed game theoretic optimization framework on profit maximization and load balancing.

Compared with the conference version [1], this manuscript has the following extensions: (i) Most importantly, we consider realistic power grid structure (for example, the IEEE 24-bus structure) in this manuscript, and analyze the robustness and stability of the power grid based on the DC power flow analysis. In this way we can achieve better analysis of power grid robustness and risk of overflow compared with the conference version due to accounting for the actual branching structures of the power grid. (ii) We consider renewable power generations in this manuscript. In the power-dependent pricing scenario in this manuscript, we set the power-dependent pricing signal at each power bus to be dependent on both load power consumption and renewable power generation, in order to incentivize the cloud computing controller to schedule workloads to power buses with more abundant renewable power generation due to reduced overall energy cost. In the time-ahead pricing scenario in this manuscript, the power grid controller will optimize the price(s) at each power bus in order to fully utilize the renewable power generations. (iii) We have added Section II on related works and more comprehensive experimental results based on two different cloud computing system specifications compared with the conference version.

II. RELATED WORK

There is a plenty of research work on the energy cost minimization of geographically distributed data centers connected to the smart power grid with location-dependent dynamic energy pricing. Most of them [26], [27] optimize workload distribution by shifting/routing computation workloads towards data center located in regions with cheaper electricity or with excessive generated electrical energy from the renewable energy sources. Moreover, reference work [28] derives the optimal control of the cloud computing system in order to enhance the power grid stability and robustness, which is different from our paper since we assume two players/agents, the power grid controller and cloud controller, in the system, and study the interaction between them.

Some research works have studied the interaction of the smart power grid with load devices using a game theoretic framework. For example, references [24], [34] propose a game theoretic framework of the smart power grid with distributed plug-in electric vehicles (PEVs), in which the power-dependent pricing scenario is applied and PEVs achieve the Nash equilibrium in battery charging/discharging in one time slot. Moreover, reference [35] proposes a Stackelberg game-based framework of the smart power grid applying time-ahead pricing with distributed PEVs in one time slot. The power grid controller is the leading player and distributed PEVs constitute the second player in this formulation. Our paper is different from these works since (i) we consider the interaction between cloud computing and smart power grid, along with other load devices and renewable power generation, (ii) we consider the case of location-dependent dynamic energy pricing at multiple power buses and adopt power flow analysis to assess grid stability and robustness, and (iii) we consider both power-dependent pricing and time-ahead pricing.

III. SYSTEM MODEL

In this section, we introduce the notation and system models for both power and data networks. We focus on a specific time slot during system operation in rest of the paper.

A. The Smart Power Grid

Consider a smart power grid and let $\mathcal{N}$ with size N denote the set of all *power buses*, indexed by integer *i*. The power buses are interconnected through *branches* forming the grid topology. Each bus $i \in \mathcal{N}$ is connected to non-renewable and/or renewable power generators and 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.

Let $P_{bus,i}$ denote the amount of *active power injection*, i.e., total power generation minus total load, at bus *i*. Let $P_{bus,i}^{Gen}$ and $P_{bus,i}^{Load}$ denote the total power generation and load power consumption at bus *i*, respectively. We have:

$$P_{bus,i} = P_{bus,i}^{Gen} - P_{bus,i}^{Load} \tag{1}$$

For the overall power grid, the total power generation and load consumption balance with each other, i.e.,

$$\sum_{i=1}^{N} P_{bus,i}^{Gen} = \sum_{i=1}^{N} P_{bus,i}^{Load} \tag{2}$$

Next we will elaborate the components of $P_{bus,i}^{Gen}$ and $P_{bus,i}^{Load}$.

Let $P_{bus,i}^{Gen,NR}$ and $P_{bus,i}^{Gen,R}$ denote the non-renewable and renewable (PV) power generations, respectively, at power bus *i*. We have:

$$P_{bus,i}^{Gen} = P_{bus,i}^{Gen,NR} + P_{bus,i}^{Gen,R} \tag{3}$$

On the other hand, the total load consumption at bus *i* is calculated via:

$$P_{bus,i}^{Load} = P_{bus,i}^{DC} + P_{bus,i}^{Back} \tag{4}$$

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

$$P_{bus,i}^{DC} = \sum_{bus(j)=i} P_{DC,j}. \tag{5}$$

If there is no datacenter connected to bus *i*, we have $P_{bus,i}^{DC} = 0$.

Power Flow Analysis:

From the perspective of power flow analysis [33], we can derive the *DC-equivalent power flow equations*:

$$P_{bus,i} = \sum_{i' \neq i} B_{ii'}(\theta_i - \theta_{i'}), \quad \forall i \in \mathcal{N} \tag{6}$$

Here, $B_{ii'}$ denotes the imaginary term in the complex value at row i and column i' of the *Y-bus* (admittance) matrix of the grid, and θ_i denotes the angle of voltage phaser at bus *i*. In power flow equations, the only variables are angles θ_i for all buses $i \in \mathcal{N}$. In practice, we select one bus as the *slack bus* with zero phaser angle. Therefore, the phaser angles at all the other buses are selected in terms of their differences with respect to the reference phaser angle in the slack bus [28][33].

Given the phaser angles θ_1, ..., θ_N obtained by solving the system of linear equations in Eqn. (6), we can calculate the active power flow over each branch (i, i') as:

$$P_{ii'} = B_{ii'}(\theta_i - \theta_{i'}) \tag{7}$$

The amount of $P_{ii'}$ values directly affects the problem of circuit overflow in a distributed power grid. That is, overflow occurs if active power at branch (i, i') reaches its maximum permitted level P^{max}. Thus, it is required to always limit $P_{ii'}$ below the level P^{max}. In summary, whether or not the circuit overflow problem occurs in a power grid depends on the grid topology, the Y-bus matrix, and the amount of active power injection at all power buses in the system.

Power-Dependent Pricing and Time-Ahead Pricing:

We consider location-dependent dynamic pricing in this paper, i.e., the power grid controller announces different prices for different power buses. We consider two different pricing scenarios: *power-dependent pricing* and *time-ahead pricing*. In the first scenario, i.e., the power-dependent pricing scenario, the power grid controller announces the price signal vector, denoted by $\boldsymbol{price}(\boldsymbol{P}_{bus}^{Load})$, based on the power consumption vector $\boldsymbol{P}_{bus}^{Load} = \{P_{bus,1}^{Load}, P_{bus,2}^{Load}, \dots, P_{bus,N}^{Load}\}$. In order to perform load balancing by incentivizing the cloud controller to shift the loads among data centers, we set the unit energy price at the i^{th} power bus as a linear function of $P_{bus,i}^{Load} - P_{bus,i}^{Gen}$, i.e.,

$$\begin{aligned} price_i(P_{bus,i}^{Load}) &= C \cdot (P_{bus,i}^{Load} - P_{bus,i}^{Gen}) + price^{\mathbf{B}} \\ &= C \cdot (P_{bus,i}^{DC} + P_{bus,i}^{Back} - P_{bus,i}^{Gen}) + price^{\mathbf{B}} \end{aligned} \tag{8}$$

where $price^{\mathbf{B}}$ and C are constant values. This power-dependent pricing scheme is similar to [31]. In this pricing scenario, (i) $P_{bus,i}^{Gen}$ is a fixed constant value, and (ii) 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. Fig. 1 (a) illustrates the power-dependent pricing scenario, in which the price at each power bus depends on the power consumption of data centers (resource allocation results as we shall see later) and other load devices.

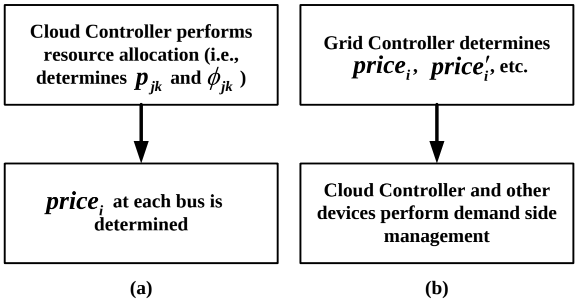


Fig. 1. Illustration of (a) the power-dependent pricing and (b) time-ahead pricing scenarios.

In the second pricing scenario, i.e., the time-ahead pricing scenario, the power grid controller announces the price signal first and the cloud computing system and other users (load devices) perform demand side management in response. In order for better performing load balancing, the power grid controller employs a *dual price* scheme similar to [29], i.e., it utilizes two potentially different unit energy prices $price_i$ and $price_i'$ for the data centers and other load devices connected to the i^{th} $(1 \le i \le N)$ power bus, respectively[1]. Intuitively, the peak power consumption of data center will cancel the trough power consumption of other load devices at each power bus under this dual-pricing policy. We also add regulations so that the average prices for data centers and for other load devices should not exceed $price_{avg,max}$ and $price_{avg,max}'$, respectively.

The cloud controller and other load devices will perform demand side management accordingly. The cloud controller determines the data center power consumption at each i^{th} $(1 \le i \le N)$ power bus, denoted by $P_{bus,i}^{DC}(\boldsymbol{price})$, based on a joint consideration of the price vector $\boldsymbol{price} = \{price_1, price_2, \dots, price_N\}$ for all the power buses, as shall be discussed later. On the other hand, the other load devices will employ distributed storage systems [23] or other load shaping techniques [30] 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.,

$$P_{bus,i}^{Back}(price_i') = P_{bus,i}^{Back}(0) - \alpha_i \cdot price_i'. \tag{9}$$

Fig. 1 (b) illustrates this time-ahead pricing scenario.

B. Resource Allocation in the Cloud Computing System

Fig. 2 shows the structure of the target resource allocation system in the cloud with a service request pool, M distributed data centers as well as a central resource management node. Each j^{th} data center is comprised of K_j potentially heterogeneous servers. We use k as the index of servers in a data center.

[1] Of course the proposed optimization framework is general enough to support the uniform-pricing scheme, i.e., applying price $price_i$ for both data centers and other load devices.

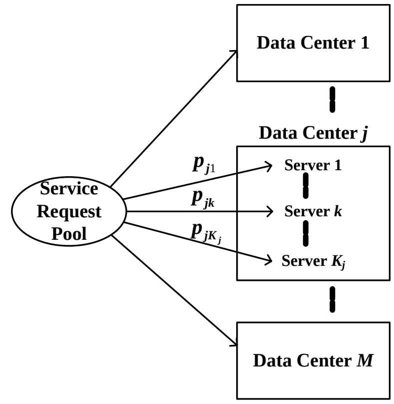


Fig. 2. Architecture of the resource allocation problem in the cloud computing system.

The service request pool contains service requests that are 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 derive the analytical form of the average response time, service requests are assumed to follow a Poisson process with an average generating rate of λ, which is 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 ϕ_{jk} $(0 \le \phi_{jk} \le 1)$, for servicing the requests. These ϕ_{jk} values are also optimization variables in the resource allocation framework. By using the well-known formula in M/M/1 queues [37], the average response time of service requests that are dispatched to that server is calculated as:

$$R_{jk}(p_{jk}, \phi_{jk}) = \begin{cases} \dfrac{1}{\phi_{jk} \cdot \mu_{jk} - p_{jk} \cdot \lambda} & \text{if } p_{jk} > 0, \\ 0 & \text{if } p_{jk} = 0, \end{cases} \tag{10}$$

where μ_{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 is comprised 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 ϕ_{jk} of the resources that have been allocated for request processing:

$$\begin{aligned} P_{Serv,jk}^{dyn}(p_{jk}) &= \frac{p_{jk} \cdot \lambda}{\phi_{jk} \cdot \mu_{jk}} \cdot \phi_{jk} \cdot P_{Serv,jk}^{dyn,max} \\ &= \frac{p_{jk} \cdot \lambda}{\mu_{jk}} \cdot P_{Serv,jk}^{dyn,max}, \end{aligned} \tag{11}$$

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 ε_{jk} and another term proportional to the portion ϕ_{jk} of allocated resources for request processing:

$$P_{Serv,jk}^{sta}(\phi_{jk}) = \varepsilon_{jk} + \phi_{jk} \cdot (P_{Serv,jk}^{sta,max} - \varepsilon_{jk}). \tag{12}$$

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

$$P_{DC,j} = \sum_{1 \le k \le K_j} \left(P_{Serv,jk}^{dyn}(p_{jk}) + P_{Serv,jk}^{sta}(\phi_{jk}) \right). \tag{13}$$

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 (in fact the profit rate) of the cloud computing system is calculated by[2]:

$$\begin{aligned} &\lambda \cdot \left(\beta - \gamma \cdot \sum_{j=1}^{M} \sum_{k=1}^{K_j} \frac{p_{jk}}{\phi_{jk} \cdot \mu_{jk} - p_{jk} \cdot \lambda} \right) - \\ &\sum_{i=1}^{N} price_i \sum_{bus(j)=i} \sum_{k=1}^{K_j} \left(P_{Serv,jk}^{dyn}(p_{jk}) + P_{Serv,jk}^{sta}(\phi_{jk}) \right). \end{aligned} \tag{14}$$

which depends on optimization variables p_{jk}'s and ϕ_{jk}'s.

IV. Optimization under Power-Dependent Pricing

We consider the interaction system of the power grid and cloud computing in the power-dependent pricing scenario, and provide the sequential game formulation comprised of two players. The cloud controller is the first player and the power grid controller is the second player as shown in Fig. 1 (a).

We know that the power grid controller (the second player) always sets the price $price_i$ as a linear function of $P_{bus,i}^{Load} - P_{bus,i}^{Gen}$ under this pricing scenario as shown in Eqn. (8), which fact is known to the cloud controller. The objective of the cloud controller is to maximize its own profit **with an anticipation of price signal vector $\boldsymbol{price} = \{price_1, price_2, \dots, price_N\}$ 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 controller are p_{jk}'s and ϕ_{jk}'s. The other parameters are either constants or functions of these control variables.

Based on the backward induction principle in Stackelberg games [38], the cloud computing controller maximizes its (anticipated) total profit given by Eqn. (14), where the anticipated price $price_i$ is a linear function of $P_{bus,i}^{Load} - P_{bus,i}^{Gen}$ as shown in Eqn. (8), and $P_{bus,i}^{Gen}$ is a fixed value. Constraints of the optimization problem include:

$$0 \le p_{jk} \le 1, \qquad \text{for } \forall j,k, \tag{15}$$

$$0 \le \phi_{jk} \le 1, \qquad for\ \forall j,k, \tag{16}$$

$$\sum_{j=1}^{M} \sum_{k=1}^{K_j} p_{jk} = 1, \tag{17}$$

$$p_{jk} \cdot \lambda < \phi_{jk} \cdot \mu_{jk}, \qquad for\ \forall j,k, \tag{18}$$

where constraints (15) and (16) specify the domains of the optimization variables. Constraint (17) ensures that all service requests can get serviced. Constraint (18) 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 optimization problem is a nonlinear programming problem and cannot be solved using conventional convex optimization methods because the objective function (14) is neither convex nor concave of optimization variables. In fact, this problem is essentially a variant of the generalized process sharing (GPS) problem discussed in [42], in which a theoretical bound in performance could be achieved compared with the actual optimal solution. Hence, we adopt 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 and an *optimal request dispatch* phase as follows.

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

Initialize the p_{jk} values.

Do the following procedure iteratively:

> Optimal resource allocation: Find the optimal ϕ_{jk} values that maximize (14) based on the derived p_{jk}'s.
>
> Optimal request dispatching: Find the optimal p_{jk} values that maximize (14) based on the derived ϕ_{jk}'s.

Until the solution converges.

The Optimal Resource Allocation Phase: In this phase, the cloud controller finds the optimal ϕ_{jk}'s in order to maximize Eqn. (14) when the p_{jk} values are given. The constraints are Eqns. (16) and (18). This problem is a convex optimization problem since the objective function (14) is a concave function of ϕ_{jk}'s when the p_{jk} values are given (please note that $price_i$ in (14) also depends on ϕ_{jk}'s), and constraints (16), (18) are linear inequality constraints. It can be optimally solved within polynomial time complexity using standard convex optimization techniques. Note that when $p_{jk} = 0$, it is possible that the optimal ϕ_{jk} value is infinitesimal. In order to find the valid ϕ_{jk} values, we add the following constraint when solving this optimization problem:

$$\phi_{jk} \ge \delta, \qquad \text{for } \forall j,k, \tag{19}$$

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

The Optimal Request Dispatch Phase: In this phase, the cloud controller finds the optimal p_{jk} values to maximize Eqn. (14) when the ϕ_{jk} values are given. The constraints are Eqns. (15), (17), and (18). This problem is also a convex

---

[2] Please note that Eqn. (10) is valid when $p_{jk} = 0$.

optimization problem since the objective function (14) is a concave function of p_{jk}'s when the ϕ_{jk} values are given in prior (please note that $price_i$ in (14) also depends on p_{jk}'s), and therefore, it could be solved optimally with polynomial time complexity using standard technique.

V. Optimization under Time-ahead Pricing

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

The objective of the power grid controller (the first player) is to achieve an optimal balance between maximizing its own profit and load balancing among power buses, **with an anticipation of the demand side managements performed by various load devices including data centers in response to price signals.** Based on the backward induction principle [38], the power grid controller aims to find the optimal dual price vectors $\boldsymbol{price} = \{price_1, price_2, \dots, price_N\}$ and $\boldsymbol{price}' = \{price'_1, price'_2, \dots, price'_N\}$. Moreover, the power grid controller will also determine the amount of non-renewable energy generations $P_{bus,i}^{Gen,NR}$'s at all power buses (please note that the amount of renewable energy generations $P_{bus,i}^{Gen,R}$'s are determined by environmental conditions and thus cannot be adjusted.)

Let $C_i\left(P_{bus,i}^{Gen,NR}\right)$ denote the cost for generating $P_{bus,i}^{Gen,NR}$ amount of non-renewable energy, which is a convex and increasing function of $P_{bus,i}^{Gen,NR}$. Then the total profit[3] (revenue - cost) of the power grid controller is given by:

$$Total\_Profit = \sum_{i=1}^{N}\left(price_i \cdot P_{bus,i}^{DC}(\boldsymbol{price}) + price'_i \cdot P_{bus,i}^{Back}(price'_i)\right) - \sum_{i=1}^{N} C_i\left(P_{bus,i}^{Gen,NR}\right) \tag{20}$$

where $P_{bus,i}^{DC}(\boldsymbol{price})$ and $P_{bus,i}^{Back}(price'_i)$ $(1 \le i \le N)$ are the (anticipated) power consumption values after the load devices (including data centers) have performed demand side managements based on the dual price vectors. We consider two possible objective functions of the power grid controller. In the first case, the power grid controller maximizes the following objective function:

$$w_1 \cdot Total\_Profit - w_2 \cdot \max_{i,i'} P_{ii'} \tag{21}$$

where $\max_{i,i'} P_{ii'}$ denotes the (anticipated) maximum amount of power flowing over branch (i, i'). A larger value in $\max_{i,i'} P_{ii'}$ indicates worse load balancing since the amount of $P_{ii'}$ directly affects the problem of circuit overflow in a power grid. In the second case, the power grid controller maximizes $Total\_Profit$, subject to some constraint on $\max_{i,i'} P_{ii'}$. 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 $\boldsymbol{price}$ and $\boldsymbol{price}'$, respectively) should not exceed $price_{avg,max}$ and $price'_{avg,max}$, respectively. Moreover, Eqn. (2) needs to be satisfied when determining the $P_{bus,i}^{Gen,NR}$ values to make the total power generation balanced with total power consumption in the power system.

We name this optimization problem performed in the power grid controller the *Optimal Pricing with Anticipation of Demand side managements* (OPAD) problem. Please note that the OPAD problem is the optimization of the grid controller's action with anticipation of what the cloud computing system and other load devices will perform given the grid controller's action, and thereby the optimization of OPAD problem is performed purely by the grid controller. We introduce an effective sub-optimal solution of the OPAD problem in the following.

Effective Sub-Optimal Solution of the OPAD Problem:

Suppose that the price vector $\boldsymbol{price}$ is announced by the power grid controller, then the objective of the cloud controller is to maximize its total profit given by Eqn. (14). The optimization variables are p_{jk}'s and ϕ_{jk}'s. This profit maximization problem is a simplified version of the RAAP problem defined in Section IV since $\boldsymbol{price}$ is given in prior. However, it is still not a convex optimization problem. We propose an iterative sub-optimal solution similar to Algorithm 1. Each iteration in the solution is comprised of an optimal resource allocation phase that finds the optimal ϕ_{jk}'s with given p_{jk} values, and an optimal request dispatch phase that finds the optimal p_{jk}'s with given ϕ_{jk} values. We solve a convex optimization with polynomial time complexity in each phase. Details are omitted due to space limitation. Based on the p_{jk} and ϕ_{jk} values obtained from the above profit maximization problem, we calculate $P_{bus,i}^{DC}(\boldsymbol{price})$ at each i^{th} $(1 \le i \le N)$ power bus using Eqns. (5), (13).

On the other hand, suppose that the other price vector $\boldsymbol{price}'$ has been announced by the power grid controller, then the power consumption of the other load devices than the data centers at each i^{th} $(1 \le i \le N)$ power bus, i.e., $P_{bus,i}^{Back}(price'_i)$, is calculated using Eqn. (9).

Since the OPAD problem is integrated with a cloud computing profit maximization problem, it is not possible to derive the analytical form of $P_{bus,i}^{DC}(\boldsymbol{price})$ as a function of the price vector $\boldsymbol{price}$. 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 the an effective sub-optimal solution of the OPAD problem. As pointed out in [43], the simulated annealing approach can converge to the optimal solution with probability one if the "temperature" in the algorithm reduces in infinite small speed. In actual implementations simulated annealing is an effective sub-optimal solution and the optimality gap can be obtained

[3] Please note that this total profit is an anticipated value calculated by the power grid controller.

from [43]. In this problem, the optimization variables are price vectors $\boldsymbol{price}$ and $\boldsymbol{price}'$, and the $P_{bus,i}^{Gen,NR}$ values. When we are optimizing objective function (21) in the first case, we adopt a two-step optimization procedure in order to minimize the computation overhead, as shown in Algorithm 2. In the first step, we optimize the price vectors $\boldsymbol{price}$ and $\boldsymbol{price}'$ and use certain heuristics to determine the $P_{bus,i}^{Gen,NR}$ values, whereas in the second step we optimize the $P_{bus,i}^{Gen,NR}$ values to maximize objective function (21). Algorithm 3 illustrates the detailed procedure of the first step as an example. In the second case when we maximize $Total\_Profit$ subject to a constraint on $\max_{i,i'} P_{ii'}$, we also adopt a two-step optimization procedure. In the first step, our focus is to make sure that the constraint on $\max_{i,i'} P_{ii'}$ is satisfied, and after the constraint is satisfied, we enter the second step to maximize $Total\_Profit$. Details are omitted due to space limitation.

Algorithm 2: Overview of the Effective Sub-Optimal Solution of the OPAD Problem in the First Case.

Step I: Optimize the price vectors $\boldsymbol{price}$ and $\boldsymbol{price}'$ and use certain heuristics to determine the $P_{bus,i}^{Gen,NR}$ values, as shown in Algorithm 3.

Step II: Optimize the $P_{bus,i}^{Gen,NR}$ values to maximize (21) also using simulated annealing.

Algorithm 3: Detailed Procedure of the First Step of the Effective Sub-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 $\boldsymbol{price}$ and $\boldsymbol{price}'$ satisfying the average price constraints.
>
> **Initialize** the p_{jk} values.
>
> **Do** the following procedure iteratively:
>
> > Optimal resource allocation: Find the optimal ϕ_{jk}'s that maximize (14) based on the derived p_{jk} values and $\boldsymbol{price}$.
> >
> > Optimal request dispatching: Find the optimal p_{jk}'s that maximize (14) based on the derived ϕ_{jk} values and $\boldsymbol{price}$.
>
> **Until** the solution converges.
>
> Calculate $P_{bus,i}^{DC}(\boldsymbol{price})$ for $1 \le i \le N$ using (5), (13), based on the derived ϕ_{jk} and p_{jk} values.
>
> Calculate $P_{bus,i}^{Back}(price_i')$ for $1 \le i \le N$ using (9).
>
> Use certain heuristics to set the $P_{bus,i}^{Gen,NR}$ values such that Eqn. (2) is satisfied.
>
> $Obj \leftarrow$ the value of the objective function (21) based on the calculated $P_{bus,i}^{DC}(\boldsymbol{price})$, $P_{bus,i}^{Back}(price_i')$, and $P_{bus,i}^{Gen,NR}$ values.
>
> **If** $Obj \ge Obj_{max}$: Accept the change of $\boldsymbol{price}$ and $\boldsymbol{price}'$.
>
> **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.

VI. EXPERIMENTAL RESULTS

In this section, we implement the interaction system of smart power grid and cloud computing and demonstrate the effectiveness of the proposed Stackelberg game-based optimization framework. We use normalized amounts of most of the parameters in the system instead of their actual values.

We test on the power grid topology from the IEEE 24-bus Reliability Test System as shown in Fig. 3, which is comprised of 24 power buses and 38 branches. Some of the power buses are equipped with non-renewable power generation facilities or electric loads (other than data centers), as noted in Fig. 3. For power buses equipped with non-renewable power generation facilities, the amounts of non-renewable power generation are assumed to be fixed in the power-dependent pricing scenario and are optimization variables in the time-ahead pricing scenario. For power buses equipped with electric loads, we assume that the parameter $P_{bus,i}^{Back}(0)$ is uniformly distributed 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 α_i parameters are set to be 6. Moreover, we add renewable power generation facilities to each power bus, and the amount of renewable power generation at each power bus is assumed to be uniformly distributed between 0 and 20. In general, we properly set the power generation and consumption values such that the renewable and non-renewable power generations, and data center and other load device power consumptions are comparable in magnitude with each other.

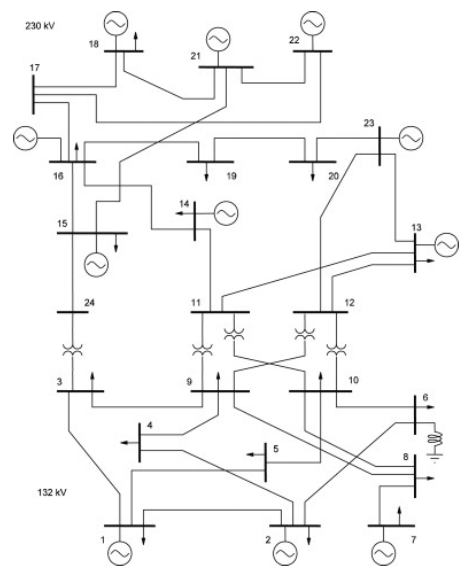


Fig. 3. Power bus topology of the IEEE 24-bus reliability test system [32].

We consider two different cloud computing environments in our evaluation, a smaller one and a larger one. The smaller cloud computing system comprises four data centers in the interaction system, comprised of 6 servers, 10 servers, 12 servers, and 20 servers, respectively. The four data centers are connected to BUS2, BUS7, BUS13, and BUS15, respectively, in the 24-bus Reliability Test System. The average service request generating rate in this cloud computing system is assumed to be 30. The larger cloud computing system comprises six data centers in the interaction system, comprised of 5 servers, 8 servers, 10 servers, 12 servers, 14 servers, and 20 servers, respectively. The six data centers are connected to BUS2, BUS7, BUS13, BUS15, BUS16, BUS18, respectively, in the 24-bus Reliability Test System. The average service request generating rate in this case is assumed to be 50. In both cloud computing systems, the maximum average service request processing rate μ_{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, parameter β is set to 6 and γ is 1.

A. Experiments under Power-Dependent Pricing

In the first experiment, we consider the interaction system under the power-dependent pricing scenario. We set the amount of non-renewable energy generation at each bus to be a fixed value 20. Also the amount of electric loads other than data centers at each power bus is assumed to be a fixed value $P_{bus,i}^{Back}(0)$. For the power-dependent pricing function, we set the base price $price^{\mathbf{B}}$ to be 0.3 and change the value of C in the experiment. Please note that parameters $price^{\mathbf{B}}$ and C are essentially relative values.

We compare the profit maximization capability of the smaller and larger cloud computing systems using the proposed Stackelberg game-based optimization method and baseline algorithm. The baseline system distributes service requests with equal probability to each server in the cloud computing system. Fig. 4 and Fig. 5 illustrate the normalized total profits of the smaller and larger cloud computing systems, respectively, versus the C value in the power-dependent pricing function. We can observe that the proposed game theoretic method consistently outperforms the baseline algorithm. When $C = 0.10$ for the larger cloud computing system, the total profit of the cloud computing system obtained by the proposed optimization method is 204.6% of that in the baseline algorithm. When $C = 0.14$ or more, the total profit in the baseline system drops below zero, and is thereby not even comparable with the proposed near-optimal method.

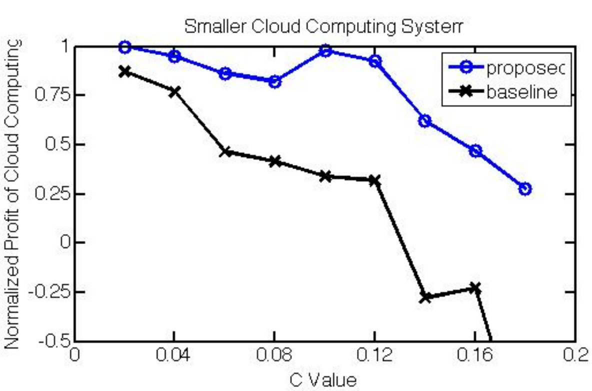


Fig. 4. The normalized total profit of the smaller cloud computing system versus the C value in the power-dependent pricing function of the proposed method and baseline algorithm.

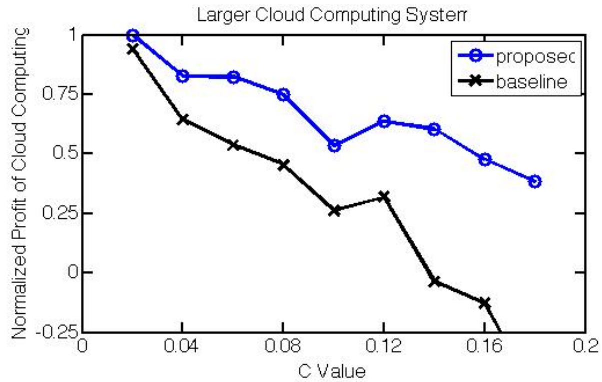


Fig. 5. The normalized total profit of the larger cloud computing system versus the C value in the power-dependent pricing function of the proposed method and baseline algorithm.

B. Experiments under Time-Ahead Pricing

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 power grid system using the proposed Stackelberg game-based optimization method and baseline algorithm. The baseline algorithm sets the same price $price_{avg,max}$ for data centers and other load devices over all power buses. We first consider the case in which the power grid controller adopts the uniform-pricing scheme, i.e., applying the same price vector $\boldsymbol{price}$ for both data centers and other load devices, and it optimizes the price vector $\boldsymbol{price}$ together with the $P_{bus,i}^{Gen,NR}$ value, in order to maximize the objective function (21). Fig. 6 illustrates the tradeoff curve (obtained by adjusting parameters w_1 and w_2) between higher profit and lower risk of circuit overflow for the power grid, assuming the case of larger cloud computing system. We can observe that simultaneous enhancement in total profit and reduction in the risk of circuit overflow (which is represented by the maximum $P_{ii'}$ value) can be achieved using the proposed Stackelberg game-based method.

Next, we consider the case in which the power grid applies the dual-pricing policy. In this case, the proposed system optimizes the price vectors $\boldsymbol{price}$ and $\boldsymbol{price}'$, and the $P_{bus,i}^{Gen,NR}$ values, in order to maximize the objective function (21). Fig. 7 and Fig. 8 illustrate the tradeoff curves (obtained by adjusting parameters w_1 and w_2) between higher profit and lower risk of circuit overflow for the power grid, assuming the cases of smaller cloud computing system and larger cloud computing system, respectively. We can observe again that simultaneous enhancement in total profit and reduction in the risk of circuit overflow can be achieved using the proposed Stackelberg game-based method. When taking an overall look of Fig. 6 to Fig. 8, we can observe that the proposed game theoretic optimization method is extremely powerful in performing load balancing, i.e., it reduces the maximum $P_{ii'}$ value in the smart power grid system by a factor up to 50X.

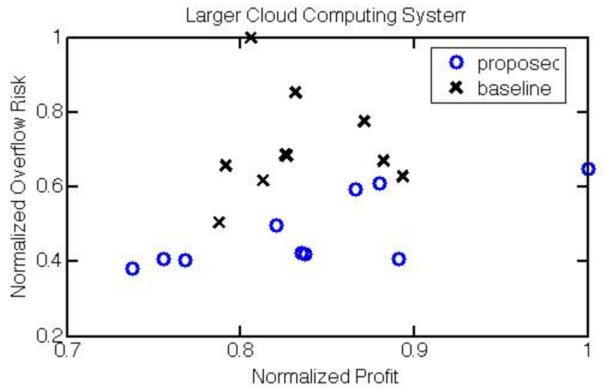


Fig. 6. The tradeoff between total profit of the smart power grid system and risk of circuit overflow in the smart power grid using uniform-pricing scheme.

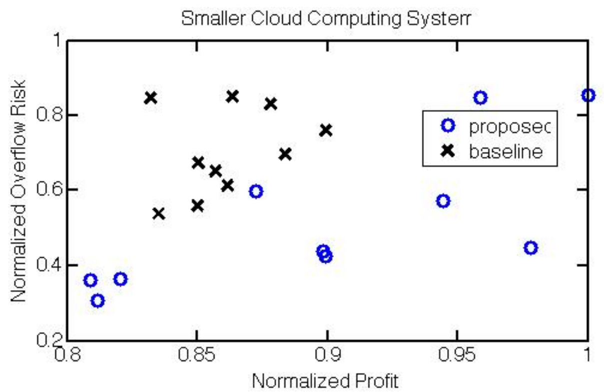


Fig. 7. The tradeoff between total profit of the smart power grid system and risk of circuit overflow in the smart power grid using dual-pricing scheme, smaller cloud computing system.

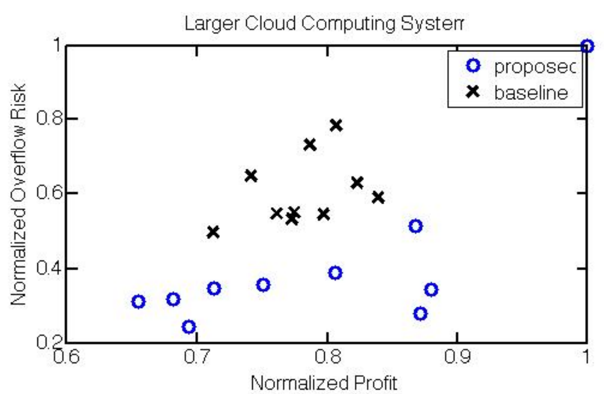


Fig. 8. The tradeoff between total profit of the smart power grid system and risk of circuit overflow in the smart power grid using dual-pricing scheme, larger cloud computing system.

VII. Conclusion

In this paper, we consider an interaction system of the smart power grid with distributed PV power generation and the cloud computing system, jointly taking into account the service request dispatch and routing problem in the cloud with the power flow analysis in power grid. The smart power grid employ dynamic pricing policies to incentivize the cloud controller to shift the computation load towards data centers located in regions with cheaper electricity. Data centers also provide opportunities to help the power grid with respect to robustness and load balancing. We provide the Stackelberg game formulation of the interaction system under two different pricing scenarios: real-time power-dependent pricing and time-ahead pricing. The two players in the Stackelberg games are the power grid controller that sets the pricing signal and the cloud controller that performs resource allocation among data centers. The objective of the power grid controller is to maximize its own profit and perform load balancing among power buses, i.e., minimizing the power flow from one power bus to the others, whereas the objective of the cloud controller is to maximize its own profit with respect to the location-dependent pricing signal. Based on the backward induction method, we derive the near-optimal or sub-optimal strategies of the two players in the Stackelberg game using convex optimization and heuristic search techniques.

VIII. References

[1] Y. Wang, X. Lin, and M. Pedram, “A sequential game perspective and optimization of the smart grid with distributed data centers,” *Proc. of IEEE PES Innovative Smart Grid Technologies Conference* (ISGT), 2013.

[2] B. Hayes, “Cloud Computing,” *Communications of the ACM*, 2008.

[3] B. Buyya, “Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility,” in *Proc. of IEEE/ACM International Symposium on Cluster Computing and the Grid* (CCGrid), 2009.

[4] M. Pedram, “Energy-efficient datacenters,” *IEEE Trans. on Computer-Aided Design*, 2012.

[5] M. Armbrusk, 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.

[6] R. H. Katz, “Tech Times Building Boon,” *IEEE Spectrum*, 2009.

[7] L. A. Barroso and U. Holzle, “The case for energy-proportional computing,” *IEEE Computer*, 2007.

[8] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," *Software Practice and Experience*, 2002.

[9] L. Zhang and D. Ardagna, "SLA based profit optimization in autonomic computing systems," in *2nd Int. Conf. on Service Oriented Computing*, 2004.

[10] D. Ardagna, M. Trubian, and L. Zhang, "SLA based resource allocation policies in autonomic environments," *Journal of Parallel and Distributed Computing*, 2007.

[11] A. Chandra, W. Gongt, and P. Shenoy, "Dynamic resource allocation for shared clusters using online measurements," *International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS), 2003.

[12] M. N. Bennani and D. A. Menasce, "Resource allocation for autonomic clusters using analytic performance models," in *Proc. of the 2nd International Conference on Autonomic Computing*, 2005.

[13] Y. Wang, S. Chen, H. Goudarzi, and M. Pedram, “Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model,” *Proc. of International Symposium on Quality Electronic Design* (ISQED), 2013.

[14] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," *Proc. of IEEE International Conference on Cloud Computing* (CLOUD), 2011.

[15] Y. Wang, S. Chen, and M. Pedram, “Service level agreement-based joint application environment assignment and resource allocation in cloud computing systems," in *IEEE Green Technologies Conference* (GreenTech), 2013.

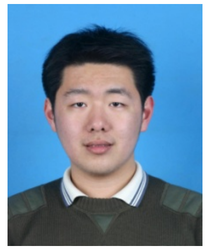
[16] Y. Gao, Y. Wang, S. K. Gupta, and M. Pedram, "An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems," in *Proc. of the International Conference on Hardware/Software Codesign and System Synthesis* (CODES+ISSS), 2013.

[17] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method for fair resource allocation for cloud computing services," *Journal of Supercomputing*, 2010.

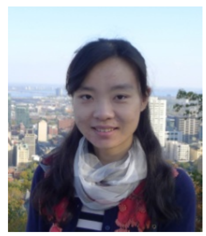
[18] United States Environmental Protection Agency, "EPA report on server and data center energy efficiency," Final Report to Congress, Aug. 2007.

[19] L. D. Kannberg, D. P. Chassin, J. G. DeSteese, S. G. Hauser, M. C. Kintner-Meyer, R. G. Pratt, L. A. Schienbein, and W. M. Warwick,

"GridWise™: The benefits of a transformed energy system," PNNL-14396, Pacific Northwest National Laboratory, Richland, Sept. 2003.
[20] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power and Energy Magazine*, 2005.
[21] S. Kishore and L. V. Snyder, "Control mechanisms for residential electricity demand in SmartGrids," *Proc. of IEEE SmartGridComm*, 2010.
[22] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the Smart Grid," *Proc. of IEEE SmartGridComm*, 2010.
[23] 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.
[24] C. Wu, H. Mohsenian-Rad, J. Huang, "Vehicle-to-Aggregator Interaction Game," *IEEE Transactions on Smart Grid*, vol. 3, No. 1, March 2012.
[25] M. Ghamkhari and A. H. Mohsenian-Rad, "Optimal integration of renewable energy resources in data centers with behind-the-meter renewable generator," *Proc. of International Conference on Communications* (ICC), 2012.
[26] 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*, San Diego, CA, Mar. 2010.
[27] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. Andrew, "Greening geographical load balancing," in *Proc. of ACM Sigmetrics*, 2011.
[28] A. H. Mohsenian-Rad and A. Leon-Garcia, "Coordination of cloud computing and smart power grids," *Proc. of IEEE SmartGridComm*, 2010.
[29] S. Yue et al., "Dual-pricing policy for controller-side strategies in demand side management", *Proc. of IEEE SmartGridComm*, 2011.
[30] T. Cui, Y. Wang, S. Yue, S. Nazarian, and M. Pedram, "An incentive-based price determination algorithm for utility companies in smart grid under game theoretic models," in *IEEE PES Innovative Smart Grid Technologies Conference* (ISGT), 2013.
[31] 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.
[32] "Reliability Test System Task Force of the Application of Probability Methods subcommittee – The IEEE Reliability Test System – 1996," pp. 1010 – 1020, Aug. 1999.
[33] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. Wiley-Interscience, 2996.
[34] C. Wu, H. Mohsenian-Rad, and J. Huang, "Wind power integration via aggregator-consumer coordination: a game theoretic approach," in *Proc. of IEEE Innovative Smart Grid Technologies* (ISGT) *Conference*, 2012.
[35] W. Tushar, W. Saad, H. V. Poor, and D. B. Smith, "Economics of electric vehicle charging: a game theoretic approach," *IEEE Trans. on Smart Grid*, 2012.
[36] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 3rd edition, 1991.
[37] L. Kleinrock, *Queueing Systems, Volume I: Theory*, New York: Wiley, 1975.
[38] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*, Morgan & Claypool Publishers, 2008.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[40] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Feb. 2011.
[41] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, No. 4598, pp. 671 – 680.
[42] A. K. Parekh, "A generalized processor sharing approach to flow control in integrated service networks," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, February 1992.
[43] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing," *Statistical Science*, 1993.



Yanzhi Wang (S'12) received the B.S. degree with distinction in electronic engineering from Tsinghua University, Beijing, China, in 2009, and Ph.D. degree in electrical engineering at University of Southern California, in 2014, under the supervision of Prof. Massoud Pedram. He is currently a postdoctoral research associate and (part-time) lecturer at University of Southern California. His current research interests include system-level power management, next-generation energy sources, hybrid electrical energy storage systems, near-threshold computing, digital circuits power minimization and timing analysis, cloud computing, mobile devices and smartphones, electric vehicles and hybrid electric vehicles, and the smart grid. He has published around 120 papers in these areas. He received best paper awards at 2014 IEEE International Symposium on VLSI (ISVLSI) and 2014 IEEE/ACM International Symposium on Low Power Electronics Design (ISLPED).



Xue Lin (S'12) received the B.S. degree from Tsinghua University, Beijing, China, in 2009. She is now a Ph.D. student in the Department of Electrical Engineering at University of Southern California. Her advisor is Prof. Massoud Pedram. She has been working on power management of photovoltaic (PV) systems, near-threshold computing of FinFET circuits, task scheduling in real-time and mobile systems, and power management of energy storage systems and electric vehicles. She has published more than 40 papers in these areas. She received best paper award at 2014 IEEE International Symposium on VLSI (ISVLSI).

Massoud Pedram (F'01), who is the Stephen and Etta Varra Professor in the Ming Hsieh department of Electrical Engineering at University of Southern California, received a Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1991. He holds 10 U.S. patents and has published four books, 13 book chapters, and more than 140 archival and 380 conference papers. His research ranges from low power electronics, energy-efficient processing, and cloud computing to photovoltaic cell power generation, energy storage, and power conversion, and from RT-level optimization of VLSI circuits to synthesis and physical design of quantum circuits. For this research, he and his students have received seven conference and two IEEE Transactions Best Paper Awards. Dr. Pedram is a recipient of the 1996 Presidential Early Career Award for Scientists and Engineers, a Fellow of the IEEE, an ACM Distinguished Scientist, and currently serves as the Editor-in-Chiefs of the ACM Transactions on Design Automation of Electronic Systems and the IEEE Journal on Emerging and Selected Topics in Circuits and Systems. He has also served on the technical program committee of a number of premiere conferences in his field and was the founding Technical Program Co-chair of the 1996 International Symposium on Low Power Electronics and Design and the Technical Program Chair of the 2002 International Symposium on Physical Design.