Bilateral Electricity Trade Between Smart Grids and Green Datacenters: Pricing Models and Performance Evaluation

Zhi Zhou, Fangming Liu*, *Senior Member, IEEE,* Zongpeng Li, *Senior Member, IEEE*

***Abstract*—Datacenter demand response is a promising approach for mitigating operational instability faced by smart grids. It enables significant potentials in peak load shedding and facilitates the incorporation of distributed generation and intermittent energy sources. This work considers two key aspects towards realtime electricity pricing for eliciting demand response: (i) Two-way electricity flow between smart grids and large datacenters with hybrid green generation capabilities. (ii) The geo-distributed nature of large cloud systems, and hence the potential competition among smart grids that serve different datacenters of the cloud. We propose a pricing scheme tailored for geo-distributed green datacenters, from a multi-leader (smart grids) single-follower (cloud) game point of view. At the cloud side, in quest for scalability, robustness and performance, the energy cost minimization problem is solved in a distributed manner, based on the technique of alternating direction method of multipliers (ADMM). At the smart grid side, a practical equilibrium of the multi-leader single-follower pricing game is desired. To this end, we employ the technique of equilibrium problem with equilibrium constraints (EPEC) and exact linearization, to accurately transform the multi-leader single-follower pricing game which is non-convex into a mixed integer linear system that can be readily solved. The effectiveness of the proposed solutions is evaluated based on real datacenter workload traces and IEEE 14-bus test systems with real generation and demand data.**

***Index Terms*—Smart grids, geo-distributed datacenters, bilateral pricing, lead-follower game.**

I. Introduction

The recent years have witnessed new technology advances in the ICT sector, two of which are of practical significance and of particular interest to this work. The first is the Internet-scale cloud services deployed over geographically distributed datacenters, serving enterprises and end users, indispensable for a wide variety of applications. The second is the evolution of the traditional power grids to smart grids, enabling sustainable, cost-effective, and environment-friendly electric power generation and consumption.

Further developments of both cloud computing and smart grids are facing their respective challenges. In smart grids, the integration of a large number of generation units (*e.g.*, wind turbines and solar arrays) incurs operation stability concerns and hence economic issues, due to the intermittent nature of such distributed generation [14]. For example, the enormous wind generation in Germany in May 2014 resulted in continuous negative electricity prices for five hours [4]. Furthermore, the soaring of peak demand exacerbates the vulnerability and price fluctuation of a power grid. These operational issues can be addressed by demand response from large loads, among whom datacenters are natural candidates. For a datacenter, the power drawn from the smart grid is often of very large volumes yet exhibiting an elastic nature. An individual datacenter can make up 50% of the load of a distribution grid nowadays [21] (*e.g.*, Facebook's datacenter in Crook County, Oregon). Despite its sheer volume, datacenter power consumption is a natural target in demand response since it is driven by user requests that can be split to geo-distributed datacenters, and be served by multiple energy sources. Thus, the power drawn by a datacenter from the smart grid can be flexibly adjusted by changing the workload routed to the former, or modulating the output of on-site generations.

In response to escalating pressure from economic and environmental concerns, many cloud providers are installing hybrid renewable energy systems that include off-site wind, solar farms and on-site fuel cells. For economies of scale, the installed capacity of wind and solar generation is often very large, as illustrated in Table I. The farms are constructed mostly at locations with desirable climate, and their realtime output often significantly exceeds datacenter demand (usually dozens of MW); *e.g.*, as one of the world's largest datacenters, the capacity of Microsoft's Chicago datacenter is 60MW [7], much smaller than the 175 MW capacity of the nearby wind farm. However, storing a large amount of excess wind or solar power in energy storage device (ESD) is rather expensive, and large capacity lead-acid ESD is not environment friendly. Fortunately, by participating in the existing *net metering* [9] program, the excess renewable power can be sold back to the smart grid, eliminating the need of expensive large capacity ESDs. Such an option is already adopted in reality, including the cloud providers in Table I [8], [1], [6], [2]. Furthermore, fuel cell generation is emerging as a reliable complement to intermittent renewable energy such as wind or solar, with predictable and controllable output levels. Compared to traditional on-site stand-by generation (*e.g.*, diesel generators), datacenter fuel cells typically run on carbon neutral direct biogas, and hence is much more environment-friendly.

Given the soaring energy cost of cloud computing, it is critical to coordinate multi-energy-resources as well as datacenters geographically distributed at different regions, to serve user demands in a cost-efficient manner. Specifically, given a performance target in terms of user-perceived latency of service, the energy cost of the cloud is expected to be minimized. This goal can be achieved via a joint optimization on *geographically load balancing*, *i.e.*, at each front-end server, determining the amount of workload distributed to each

Z. Zhou and F. Liu are with the Services Computing Technology and System Lab, Cluster and Grid Computing Lab in the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China. The corresponding author is F. Liu. E-mail: {zhiz, fmliu}@hust.edu.cn.

Z. Li is with the Department of Computer Science, University of Calgary, Canada. E-mail:zongpeng@ucalgary.ca.

TABLE I: Renewable Generation at Large Datacenters.

Company	Location	Type	Capacity	Remark
Microsoft [8]	Chicago, IL	Wind	175MW	60MW datacenter capacity [7]
Facebook [2]	Altoona, IA	Wind	138MW	100% wind powered datacenter
Google [6]	Mayes, OK	Wind	48MW	
Apple [1]	Maiden, NC	Solar	40MW	10MW additional fuel cells
Google [6]	Council Bluffs, IA	Wind	114MW	
Microsoft [8]	San Antonio, TX	Wind	110MW	

datacenter; and *green capacity planning*, *i.e.*, determining the amount of fuel cell generation, power drawn from and sold back to the corresponding smart grid. Note that geographical load balancing leverages the spatial diversity of energy prices across datacenters, while green capacity planning takes advantage of the price differences of multi-energy-resources within a datacenter, thus a joint optimization on these two knobs efficiently reduce the energy cost of the cloud.

Datacenter with hybrid energy supplies can provide great potential to demand response, by setting the appropriate amount of energy drawn from or sold back to the smart grid. Unfortunately, for a smart grid, despite the fact that datacenter demand response can be obtained by adjusting the electricity and/or net metering price in a real-time manner, eliciting a desired amount of demand response from the hybrid green datacenter is of great challenge. In particular, when a cloud runs on top of geo-distributed datacenters and couples multiple smart grids, a smart grid needs not only to anticipate the cloud's response to the price decisions, but also to consider the impact of other smart grids' price decisions on the cloud's response. In such competition scenarios, the price strategy of one smart grid can influence the demand response at another smart grid. Consequently, the pricing process becomes a non-cooperative game. If we further consider the cloud's response to the prices, the entire process can then be captured by a *multi-leader single-follower game* [20], in which each smart grid acts as a leader and sets the prices, while the cloud acts as the follower and responses to the prices by minimizing its energy cost.

At the cloud side, to handle upgrades (*e.g.*, installation of new datacenters) and abrupt changes in system conditions (*e.g.*, datacenter outages), the energy cost of the cloud is expected to be minimized in a distributed manner to offer better scalability, robustness and performance. To this end, alternating direction method of multipliers (ADMM) [12] is applied to decompose the global cost minimization problem into a set of small-scale subproblems that can be efficiently solved by the facilities (front-end servers and datacenters) locally. At the smart grid side, detecting an equilibrium of the bilateral pricing game is particularly challenging, since each leader's strategy space is non-convex. To address this challenge, the methods of equilibrium problem with equilibrium constraint (EPEC) [15] and exact linearization are employed to transform the non-convex system into a mixed-integer linear program (MILP) which can be readily solved.

To summarize, this work makes the following main contributions:

- We initiate the study of bilateral electricity trade between smart grids and geo-distributed datacenters, by modeling the pricing process as a multi-leader single-follower game.
- For the cloud aiming to minimize energy cost, we present an ADMM-based distributed algorithm to optimize the geo-graphical load balancing and green capacity planing.
- For the smart grids interested in the price equilibrium, the non-convex multi-leader single-follower game is transformed into a MILP that can be readily solved.
- We evaluate the efficacy of the proposed solutions based on real datacenter workload traces and IEEE 14-bus test systems with real generation and demand data.

In the rest of the paper, we discuss related work in Sec. II, and formulate the multi-leader single-follower game in Sec. III. Sec. IV and Sec. V solve the cost minimization problem at the cloud side and the smart grid side, respectively. Trace-driven simulations are presented in Sec. VI. Sec. VII discusses the limitations and the future work, Sec. VIII concludes the paper.

II. Related Work

Datacenter demand response lies at the intersection of two research areas: energy efficiency in the datacenter and demand response in the smart grid. A series of recent research was devoted to exploiting the opportunities and addressing the challenges in datacenter demand response.

Efficiently powering datacenters with green energy have been extensively studied in the literature. For example, Chen *et al.* [13] presented an online and distributed algorithm for joint workload and energy management, aiming to minimize the energy cost of geo-distributed datacenters with green energy supply. Ibrahim *et al.* [17] investigated the effect of intermittency of renewable energy availability, on a distributed storage system whose file retrieval efficiency and node repair time are optimized. When considering multiple energy resources as well as the net metering for a single datacenter, Fan *et al.* [19] proposed an framework to strike a judicious balance among energy cost, carbon emission and service availability. Different from the above work, our study further considers the performance requirement (*i.e.*, user perceived latency) at each front-end server, the coupling effect of this constraint makes our distributed algorithm design rather challenging. A variant of ADMM, ADM-G, was applied to maximize utility of geo-distributed datacenters equipped with fuel cells [35]; although directly applicable to our problem, it has worse computational efficiency. Instead, we first transform our problem to a clearer form and then apply the light-weight ADMM to improve the computational efficiency.

The market design for datacenter demand response has attracted substantial attention. In terms of lead-follower game (*a.k.a.* stackelberg game) theory, the concept is not new and has been applied to datacenter demand response. For a

colocation datacenter with multiple tenants, Tran *et al.* [27] applied a single-leader multi-follower game approach. For geo-distributed datacenters served by cooperative smarts grids, Wang *et al.* [28] proposed to balance the electricity load among those smart grids, based on a single-leader single-follower pricing game. For geo-distributed datacenters served by competing smarts grids, load reduction at each smart grid is considered in [29] and [26], through a multi-leader single-follower pricing game. Our work is different from and complementary to the above efforts in at least two important aspects: (1) we explore the potential of datacenter demand response by considering the capability of two-way electricity flow rather than one-way power consumption, and (2) we develop a more general approach to detect an equilibrium by blending EPEC and linearization techniques, since our Stage-I problem does not admit a closed-form solution and thus the backward induction method applied in the above studies is not applicable to our problem. In addition to lead-follower game theoretic approaches, Liu *et al.* [21] designed a prediction-based pricing rule for demand response of independent datacenters in one smart grid, and auction solutions have been proposed to elicit demand response from a colocation datacenter [32] and geo-distributed datacenters [37].

The basic idea of this work has been partially presented in an earlier 2-page version [36]. In the current paper, we further present the solution details that include the improved models, techniques and performance evaluations.

III. The Demand Response Model

A. Overview of the Geo-distributed Cloud Platform

Consider a cloud provider running cloud services on a set of N geographically dispersed datacenters $\mathcal{D} = \{1, 2, ..., N\}$ that are inter-connected by internet backbones. Each datacenter $j \in \mathcal{D}$ consists of S_j homogeneous processing servers. The cloud deploys a set of M front-end servers, $\mathcal{S} = \{1, 2, ..., M\}$, in a set of geographical regions, to direct user requests to appropriate datacenters.

Following recent literature on datacenter modelling [31], [37], [38], we adopt a discrete time-slotted model where the bilateral electricity prices and the decision of the cloud are updated during each time slot. In practice, real-time electricity price is often updated at the time scale of every 15 minutes or every hour. At each time slot $t = (0, 1, 2, \cdots)$, the total amount of incoming workload driven by user requests (in number of processing servers required) at the front-end server i is denoted as $D_i(t)$, and the amount of workload distributed from front-end server i to datacenter j is $d_{ij}(t)$. In the short-term (*e.g.*, 15 minutes or one hour), $D_i(t)$ can be predicted rather accurately, by employing techniques such as statistical machine learning and time series analysis [31]. Here we consider interactive workloads (*e.g.*, web search and online gaming) that can not be *deferred* and must be served *immediately*. Thus, the decision making of both the smart grids and the cloud is uncoupled from slot to slot, and we can focus on a single time slot and drop index t for the analysis henceforth.

B. Hybrid Renewable Energy Ecosystem

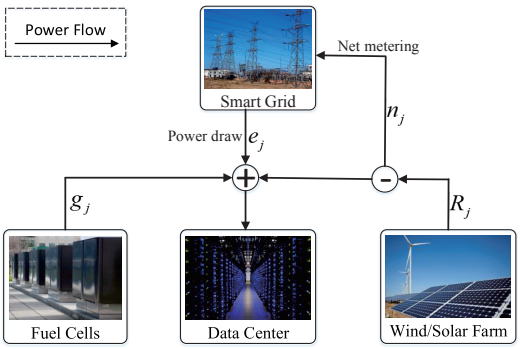

Fig. 1: A hybrid green datacenter consisting of multiple energy sources: large-scale wind/solar farms, carbon-free fuel cell generation, and power drawn from a smart grid.

Fig. 1 is an overview of a green datacenter powered by a hybrid renewable energy system that consists of multiple power supplies, including off-site intermittent wind or solar power, the emerging reliable and controllable fuel cell generation that runs on direct biogas, as well as power drawn from the macro electricity grid. Different energy sources have distinct characteristics in terms of economics, emission and reliability, and can be jointly configured towards a specific system-wide optimization goal.

Large-scale wind or solar farms: At modern cloud-scale datacenters, in quest of stability, economies of scale and datacenter extendibility, the installed capacity of wind or solar power can be very large, substantially exceeding realtime demand of a datacenter. However, the energy efficiency that determines the real output of wind and solar power is closely tied to ambient weather conditions. Consequently, wind and solar farms are often placed at locations of desirable wind speeds or solar irradiation, while not too far away from the datacenter (up to tens of miles). Under good weather conditions, the excess renewable energy can be of a large volume. However, it is impractical and not so environment-friendly to store the excess renewable energy in storage devices. Large scale electricity storage devices are still expensive; furthermore, the commonly adopted lead-acid batteries are based on polluting materials.

By participating in a program called *net metering* [9], excess renewable energy can be sold back to the electricity grid, while the unreleased Renewable Energy Certificates (RECs) [1] can still be used to offset the carbon footprint incurred by brown power drawn from the electricity grid. Therefore, when participating in net metering, though the datacenter still draws brown power from the grid, datacenter carbon-neutrality is achievable via keeping the RECs. Currently, this is a common practice of internet giants such as Google and Facebook. Constructing renewable power generation facilities (wind or solar farms) is usually expensive, but operational cost thereafter is relatively low. We hereafter use R_j to denote the output of the large-scale renewable farm owned

[1] A Renewable Energy Certificate, *a.k.a.* a "green certificate" or a "renewable energy credit", represents the generation of one MWh of electricity from an eligible renewable source. RECs are tradable commodities in REC markets, sold separately from underlying physical electricity.

by datacenter $j \in \mathcal{D}$, and use n_j to represent the amount of excess renewable power sold back to the smart grid via the net-metering program, at price p_j^{n}.

Fuel cell generation with carbon neutral biogas: A latest focal point in renewable energy is fuel cell generation from biogas. Recent technical advances have made fuel cells cost competitive. Providers such as Google, Microsoft, Apple, eBay and Facebook are investing in fuel cells to "clean up" their cloud services. Fuel cells can provide extremely reliable and high-quality power, with fully controllable output levels. Furthermore, it runs on carbon neutral direct biogas, with no environmental pressure on service providers. The cost of per-unit fuel cell generation is stable, without dramatic fluctuations over time. For datacenters, fuel cell generation also eliminates the need of traditional backup equipment such as diesel generators, UPS, batteries, and complex switchgear. This suite of legacy components is still too expensive, unreliable, and unclean. Fuel cell generation enables datacenter operators to rely on a simpler and cleaner solution. We use p_g to denote the unit cost of fuel cell generation, g_j and G_j to denote the real output and capacity of fuel cells deployed at datacenter $j \in \mathcal{D}$, respectively.

Power drawn from the smart grid: Though the installed capacity of wind or solar farms is large, and the power output of fuel cell generation is reliable and controllable, cloud-scale datacenters still can not be fully decoupled from the electricity grid — realtime wind and solar output is volatile and low at times, and the capacity of fuel cells is still limited by the current low production of direct biogas. While one may argue that brown power drawn from the electricity grid increases the carbon footprint of a datacenter, such footprint can be offset by the excess RECs produced by the large-scale wind and solar farms. We use e_j to denote power drawn from the smart grid at datacenter $j \in \mathcal{D}$, and p_j^{e} to denote the electricity price offered by the smart grid corresponding to datacenter $j \in \mathcal{D}$.

C. Datacenter Cost Minimization in Stage-II

For large-scale geo-distributed datacenters, given the escalating power demand and huge capital investment on renewable energy infrastructure (*e.g.*, wind farm, solar farm, and fuel cells), it is critical to curb the rising energy cost and utilize the renewable energy in a cost-effective manner. This goal can be achieved via the following two approaches: (1) *Geographical load balancing*, cloud operator can exploit the spacial diversity of electricity price and net metering price to reduce the electricity bill or to increase the revenue of net metering, by splitting the workload across the geo-distributed datacenters. Due to the geo-distributed feature, different datacenters usually exhibit diverse user-perceived latencies, energy costs and even carbon emission, and thus geographical load balancing determines the global latency, cost and emission at the cloud level. Unfortunately, those objectives may conflict and cannot be optimized simultaneously when performing geographical load balancing, and recent studies such as [16] have demonstrated how geo-graphical load balancing tunes the potential tradeoff among those objectives. Note that different from conventional electricity end-users such as factories and buildings that provide demand response by shifting workload in the temporal domain, for a cloud running on geo-distributed data centres, the capability of geographical load balancing further extends datacenter demand response to the spatial domain. (2) *Green capacity planning*, at each datacenter, exploiting the price diversity of different energy sources and selecting the optimal amount of power from each of them.

With recent advances in datacenter efficiency, power consumption at a datacenter can be shaped to be proportional to the amount of workload processed, by dynamically turning down idle servers and adjusting the voltage and frequency of the processors [38]. For each datacenter $j \in \mathcal{D}$, given the amount of in-coming workload $\sum_{i\in\mathcal{S}} d_{ij}$, the power demand can be denoted as $\alpha_j \sum_{i\in\mathcal{S}} d_{ij}$, where α_j represents the energy usage efficiency and transforms the amount of workload to power consumption. Here we assume that server-related energy consumption dominates the energy consumption of the cloud system, which is realistic for most cloud services such as web-search and social networking. Note that for some services such as video streaming that require few server resource but incur huge amounts of traffic and power consumption inside the datacenter LAN and the core WAN, our model is not directly applicable.

Next, for each datacenter $j \in \mathcal{D}$, given electricity cost at the smart grid, $e_j p_j^{\mathrm{e}}$, cost of fuel cell generation, $g_j p_g$, and revenue from the net metering program, $n_j p_j^{\mathrm{n}}$, the aggregated energy cost of the geo-distributed cloud is $\sum_{j\in\mathcal{D}} \left\{ e_j p_j^{\mathrm{e}} + g_j p_g - n_j p_j^{\mathrm{n}} \right\}$, which is to be minimized. However, such minimization should avoid significantly degrading the performance of cloud services. We aim to strike a judicious balance between cost control and cloud quality assurance.

Performance of Cloud Service. For interactive cloud applications such as web search and social networking, latency is a critical performance metric. We focus on the end-to-end request latency from a front-end proxy server to a processing datacenter in a wide-area network, which largely accounts for user-perceived latency and outweighs other factors such as queuing and processing delays at datacenters [31]. We assume that our distributed system is connected by a private backbone, as seen in real-world geo-distributed datacenters [23], such as Google's B4. The round-trip times (RTTs) within large datacenters with tens of thousands of servers are typically $200 - 500\mu$s [25]. At the same time, based on advances in operation systems and hardware, intra-datacenter RTTs can be as low as 1μs [25]. However, in a sharp contrast, the wide-area network latency that involves propagation, queuing, transmission, and nodal processing times in a geo-distributed system is far longer, typically tens or hundreds of milliseconds.

The RTT L_{ij} between the front-end server i and datacenter j can be obtained through active measurements [23]. Empirical studies have also demonstrated that, in backbone networks, L_{ij} can be approximated by geographical distance t_{ij} between the front-end server i and datacenter j as: $L_{ij} = t_{ij} \times 0.02$ms/km [23]. The experienced average network latency of the front-end proxy server i can be formulated as $\sum_{j\in\mathcal{D}} d_{ij} L_{ij} / D_i$. To provide satisfiable experience to end users, the user-perceived latency is generally enforced within

a given bound: $\sum_{j\in\mathcal{D}} d_{ij}L_{ij}/D_i \leq W_i$, where W_i is the maximal tolerable response delay for front-end proxy server i.

In the datacenter demand response program, given electricity price p_j^{e} and net metering price p_j^{n} at each datacenter $j \in \mathcal{D}$, the cloud minimizes the aggregated energy cost while guaranteeing its performance, by solving the following *geographical load balancing and green capacity planning problem* (GLB-GCP). Here we do not consider the carbon emission of the power drawn from the grid, since it can be fully offset by unreleased RECs and thus carbon-neutrality can be maintained, as we have explained in Sec. III-A.

GLB-GCP :

$$\min \quad \sum_{j\in\mathcal{D}} \left\{ e_j p_j^{\mathrm{e}} + g_j p_g - n_j p_j^{\mathrm{n}} \right\},$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{D}} d_{ij} = D_i, \forall i \in \mathcal{S}, \tag{1a}$$

$$\sum_{i\in\mathcal{S}} d_{ij} \leq S_j, \forall j \in \mathcal{D}, \tag{1b}$$

$$\sum_{j\in\mathcal{D}} d_{ij}L_{ij}/D_i \leq W_i, \forall i \in \mathcal{S}, \tag{1c}$$

$$e_j + g_j + R_j - n_j = \alpha_j \sum_{i\in\mathcal{S}} d_{ij}, \forall j \in \mathcal{D}, \tag{1d}$$

$$e_j \geq 0, 0 \leq g_j \leq G_j, 0 \leq n_j \leq R_j, \forall j \in \mathcal{D}, \tag{1e}$$

$$d_{ij} \geq 0, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}. \tag{1f}$$

(1a) is the workload conservation constraint that ensures all requests are served. (1b) is the datacenter capacity constraint that prevents the processed workload from exceeding datacenter capacity. (1c) is the aforementioned performance constraint ensuring that average user-perceived latency is within a tolerable range. (1d) is the power balance constraint, which requires that power demand equal to power supply. (1e) indicates that the output of fuel cell generation and the renewable power sold back to the smart grid do not exceed the installed fuel cell generation capacity and the output of renewable farm, respectively. Electricity loss in the transportation of off-site renewable energy has not been incorporated into the optimization framework, as off-site wind or solar farms are typically placed not too far away from the datacenter (up to tens of miles), in which case the loss is relatively small.

D. Non-cooperative Pricing Game in Stage-I

Before the cloud minimizes its aggregated cost by solving GLB-GCP in Stage-II, the smart grid corresponding to each datacenter j determines the realtime electricity price p_j^{e} and net metering price p_j^{n}, and announces them to datacenter j in advance in Stage-I. Changes in one smart grid's strategy ripple to other smart grids. The process of determining the electricity price and the net metering price can be modeled as a non-cooperative game.

To formulate the non-cooperative pricing game among the smart grids, we assume that a single datacenter locates within the geographical span of each regional smart grid, in line with the fact that a smart grid often covers a moderate-sized district. For the case of multiple datacenters locating within one smart grid, or one datacenter trading to multiple smart grids, the underlying problem does not change fundamentally, and we can formulate a multi-leader single-follower game in which multiple smart grids strategically trade with the geo-distributed cloud.

At the beginning of each time slot, each regional smart grid corresponding to datacenter j first computes the demand response target, *i.e.,* the desired power drawn and net metering by datacenter j, E_j and N_j, respectively, which minimize the voltage violation frequency of that smart grid. In practice, when given the structure and profile of the power distribution network (*e.g.*, the SCE 47-bus network in [21], and the IEEE 14-bus network used in performance evaluation in Sec. VI), the demand response target can be obtained by applying the "branch flow" model [21] and exploring the feasible region of power draw e_j, net metering n_j.

As we have explained in Sec. III-B, the off-site renewable facility is generally built at a distance to the datacenter and located at a different node of the smart grid, thus the smart grid concerns the demand response targets at the datacenter side and the renewable facility side separately. The difference between the power consumption target E_j and the actual power consumption incurs an economic penalty, *i.e.*, *maintenance cost caused by potential voltage violation, or cost incurred by primary frequency control for realtime load-generation balancing*, captured by a cost function $h_j^{\mathrm{e}}(E_j - e_j)$. We assume that $h_j^{\mathrm{e}}(\cdot)$ is convex, non-negative and has a global minimum $h_j^{\mathrm{e}}(0) = 0, \forall j \in \mathcal{D}$. The widely-adopted concavity assumption [21] captures increasing marginal maintenance cost in practice. Equivalently, the difference between the net metering target N_j and the actual value n_j also incurs a similar economic penalty $h_j^{\mathrm{n}}(N_j - n_j)$. We assume a quadratic cost function as follows [21]:

$$h_j^{\mathrm{e}}(E_j - e_j) = \beta_j^{\mathrm{e}}(E_j - e_j)^2,$$
$$h_j^{\mathrm{n}}(N_j - n_j) = \beta_j^{\mathrm{n}}(N_j - n_j)^2.$$

Here we adopt different penalty parameters β_j^{e} and β_j^{n} for the differences $E_j - e_j$ and $N_j - n_j$, respectively, as the nodes connected to the datacenter and off-site renewable farm may have different importance to the stability of the smart grid. The above penalty can be reduced through datacenter demand response. However, incentivizing datacenter demand response in a realtime pricing manner impacts the smart grids' profit (cost) of selling (buying) electricity to (from) the datacenter. Given the unit electricity cost P_j^{e} and original net-metering price P_j^{n}, the profit of selling electricity to datacenter j can be denoted as $e_j(p_j^{\mathrm{e}} - P_j^{\mathrm{e}})$, while the cost change of buying electricity from datacenter j can be denoted as $n_j(p_j^{\mathrm{n}} - P_j^{\mathrm{n}})$.

Then, each smart grid corresponding to datacenter j determines p_j^{e} and p_j^{n} by minimizing the total cost as follows:

$$\mathsf{SG}_j : \quad \min \quad \beta_j^{\mathrm{e}}(E_j - e_j)^2 + \beta_j^{\mathrm{n}}(N_j - n_j)^2 + e_j(P_j^{\mathrm{e}} - p_j^{\mathrm{e}}) + n_j(p_j^{\mathrm{n}} - P_j^{\mathrm{n}}), \tag{2a}$$

$$\text{s.t.} \quad P_j^{\mathrm{e}\text{-}} \leq p_j^{\mathrm{e}} \leq P_j^{\mathrm{e}+}, \tag{2b}$$

$$P_j^{\mathrm{n}\text{-}} \leq p_j^{\mathrm{n}} \leq P_j^{\mathrm{n}+}. \tag{2c}$$

Here $[P_j^{\mathrm{e}\text{-}}, P_j^{\mathrm{e}\text{-}}]$ and $[P_j^{\mathrm{n}\text{-}}, P_j^{\mathrm{n}\text{-}}]$ are feasible ranges of the electricity price and net metering price, respectively. Note that e_j and n_j above are not constants. They are determined by the datacenter cost minimization problem GLB-GCP latter in Stage-II, and thus depend on the price decisions of other smart grids. The process of determining p_j^{e} and p_j^{n} is a non-cooperative game.

IV. Distributed Geographical Load Balancing and Green Capacity Planning

The linear optimization problem GLB-GCP is of large-scale, since the number of datacenters and front-end servers can be at the scale of 10 and 10^5, respectively [31]. While it may be periodically solved at a central server, such a centralized design makes the system less responsive to system upgrade (*e.g.*, installation of new datacenters) and sudden changes in system conditions (*e.g.* link failures, datacenter outages). We design a fully distributed solution instead, for better scalability, robustness, performance, and to make use of the redundant computing resources at each datacenter and front-end server.

A classic approach to distributed algorithm design is the dual decomposition method, which relaxes selected primal constraints and then employ dual decomposition to break the problem into many sub-problems. However, dual decomposition requires the objective function to be strictly convex, the linear objective function of GLB-GCP rules out the application of dual decomposition based approachs.

In this work, we adopt the *alternating direction method of multipliers* (ADMM) [12], which overcomes the drawbacks of dual decomposition and offers fast convergence with mild assumptions on the objective function. By decoupling the green capacity planning subproblem at each datacenter from the global problem GLB-GCP, we are able to derive the former's closed-form solution. After substituting this solution at each datacenter into GLB-GCP, we take ADMM to solve the remaining geographical load balancing problem in a distributed manner.

A. Green Capacity Planning per Datacenter

Through inspecting the original problem GLB-GCP, we can find that if the total workload at each datacenter ($\sum_{i\in\mathcal{S}} d_{ij}$) is known, then each datacenter only needs to solve the following green capacity planning subproblem GCP *locally*.

$$\begin{aligned} \text{GCP}: \quad \min \quad & e_j p_j^{\text{e}} + g_j p_g - n_j p_j^{\text{n}}, \\ \text{s.t.} \quad & e_j + g_j - n_j = \alpha_j \sum_{i\in\mathcal{S}} d_{ij} - R_j, \forall j \in \mathcal{D}, \\ & e_j \geq 0, 0 \leq g_j \leq G_j, 0 \leq n_j \leq R_j, \forall j \in \mathcal{D}. \end{aligned}$$

Insight: The problem GCP admits a natural economic interpretation: given the total power demand $\alpha_j \sum_{i\in\mathcal{S}} d_{ij}$ at each datacenter $j \in \mathcal{D}$, how much power to choose from each source, for minimizing the aggregated energy cost.

An intuitive solution is to always choose the cheapest power until it is exhausted. For example, if net metering price p_j^n is cheaper than fuel cell generation price p_g and electricity price p_j^e, then we first choose the off-site renewable energy. While if the electricity price p_j^e is cheaper than net metering price p_j^n and fuel cell generation price p_g, then we should always choose the grid electricity to power the datacenter, and sell all the off-site renewable energy back to the smart grid. Furthermore, to ensure that GCP has a unique solution, we assume that when two of the three power sources have the same price, the datacenter makes power source selection in the following order: grid power first, wind and solar power second, and finally fuel cell generation.

The above intuition inspires the solution of GCP. Considering different price combinations, we list the optimal solution to GCP in Table II. With Table II, we can further obtain the aggregated energy cost at each datacenter j (denote as $C_j(\sum_{i\in\mathcal{S}} d_{ij})$). Furthermore, the relation between $C_j(*)$ and $\sum_{i\in\mathcal{S}} d_{ij}$ is depicted in Fig. 2.

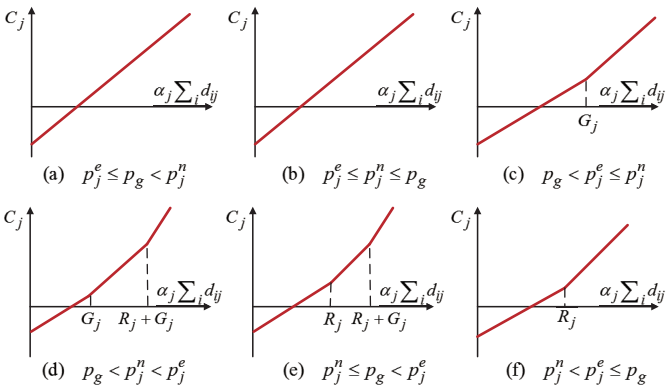

Fig. 2: The relation between the aggregated energy cost $C_j(\sum_{i\in\mathcal{S}} d_{ij})$ and the total power demand $\alpha_j \sum_{i\in\mathcal{S}} d_{ij}$, under different combinations of prices.

We can see that regardless of the price relation, the aggregated energy cost $C_j(\sum_{i\in\mathcal{S}} d_{ij})$ is always convex on d_{ij}. An intuitive explanation is that the marginal aggregated energy cost (*i.e.*, the first order derivative of $C_j(\sum_{i\in\mathcal{S}} d_{ij})$) is non-decreasing, since the optimal solution of problem GCP always chooses the cheapest energy sources first.

B. Distributed Geographical Load Balancing

We now substitute the solution to the problem GCP into the original problem GLB-GCP, then the problem GLB-GCP that minimizes the total cost of the cloud is now casted as the following master geographical load balancing problem GLB.

$$\begin{aligned} \text{GLB}: \quad \min \quad & \sum_{j\in\mathcal{D}} C_j\Big(\sum_{i\in\mathcal{S}} d_{ij}\Big), \\ \text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \forall i \in \mathcal{S}, && \text{(3a)} \\ & \sum_{i\in\mathcal{S}} d_{ij} \leq S_j, \forall j \in \mathcal{D}, && \text{(3b)} \\ & \sum_{j\in\mathcal{D}} d_{ij} L_{ij}/D_i \leq W_i, \forall i \in \mathcal{S}, && \text{(3c)} \\ & d_{ij} \geq 0, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}. && \text{(3d)} \end{aligned}$$

Given the convexity of $C_j(*)$, we propose a new distributed algorithm to solve the master GLB efficiently based on ADMM [12], a simple yet powerful algorithm for convex optimization that witnessed successful applications in image processing, machine learning and applied statistics. ADMM works well for linearly constrained convex problems whose objective function is separable into two individual convex functions with non-overlapping variables. It alternatively optimizes part of the objective with one block of variables to reach the optimum with fast convergence.

Given a convex optimization problem in the form:

$$\text{ADMM}: \Big\{ \min \ f(x) + h(z), \quad \text{s.t. } Ax + Bz = c | x \in K_1, z \in K_2 \Big\},$$

TABLE II: Solution of green capacity planning for each datacenter, given the solution of geographical load balancing.

Price Comparison	**Grid Power** (e_j)	**Fuel-cell Generation** (g_j)	**Net Metering** (n_j)
$p_j^{\mathrm{e}} \le p_g < p_j^{\mathrm{n}}$	$\alpha\sum_i d_{ij}$	0	R_j
$p_j^{\mathrm{e}} \le p_j^{\mathrm{n}} \le p_g$	$\alpha\sum_i d_{ij}$	0	R_j
$p_g < p_j^{\mathrm{e}} \le p_j^{\mathrm{n}}$	$\max\{0, \alpha\sum_i d_{ij} - G_j\}$	$\min\{G_j, \alpha\sum_i d_{ij}\}$	R_j
$p_g < p_j^{\mathrm{n}} < p_j^{\mathrm{e}}$	$\max\{0, \alpha\sum_i d_{ij} - R_j - G_j\}$	$\min\{G_j, \alpha\sum_i d_{ij}\}$	$\max\{0, R_j - \max\{0, \alpha\sum_i d_{ij} - G_j\}\}$
$p_j^{\mathrm{n}} \le p_g < p_j^{\mathrm{e}}$	$\max\{0, \alpha\sum_i d_{ij} - R_j - G_j\}$	$\min\{G_j, \max\{0, \alpha\sum_i d_{ij} - R_j\}\}$	$\max\{R_j - \alpha\sum_i d_{ij}, 0\}$
$p_j^{\mathrm{n}} < p_j^{\mathrm{e}} \le p_g$	$\max\{\alpha\sum_i d_{ij} - R_j, 0\}$	0	$\max\{R_j - \alpha\sum_i d_{ij}, 0\}$

with variables $x \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$, where $A \in \mathbb{R}^{l\times m}$ and $B \in \mathbb{R}^{l\times n}$ are relation matrices, $c \in \mathbb{R}^l$ is a relation vector. $f : \mathbb{R}^m \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$ are convex functions, and K_1, K_2 are non-empty polyhedral sets. Functions f and h are not required to be strictly convex.

The augmented Lagrangian of ADMM above can be formed by introducing an extra L_2 norm term $\|Ax + Bz - c\|_2^2$ to the objective:

$$\begin{aligned}\mathcal{L}_\rho(x, z, y) \quad = \quad & f(x) + h(z) + y^T(Ax + Bz - c) \\ & +(\rho/2)\|Ax + Bz - c\|_2^2, \end{aligned} \tag{4}$$

which can be viewed as the unaugmented Lagrangian with an extra penalty term $\rho > 0$. Consequently the minimization of $\mathcal{L}_\rho(x, z, y)$ is equivalent to the original ADMM. The quadratic penalty term is introduced to ensure strong convexity of $\mathcal{L}_\rho$, even if f and h are affine. In return, this design eliminates strong assumptions on f and g, and further substantially improves the convergence of ADMM.

At each iteration $t + 1$, the ADMM algorithm updates the original variables x, z and dual variable y in an alternating fashion:

$$\begin{aligned} x^{t+1} &= \operatorname*{argmin}_{x\in K_1} \mathcal{L}_\rho(x, z^t, y^t), \\ z^{t+1} &= \operatorname*{argmin}_{z\in K_2} \mathcal{L}_\rho(x^{t+1}, z, y^t), \\ y^{t+1} &= y^t + \rho(Ax^{t+1} + Bz^{t+1} - c). \end{aligned}$$

However, it is challenging to solve the simplified problem GLB in a distributed manner, since the quadratic penalty terms in the augmented Lagrangian of GLB is not directly decomposable. The constraints (3a)–(3c) couple the variables d_{ij} in two dimensions: the per-front-end workload conservation constraints (3a) and the performance constraint (3c) couple d_{ij} across datacenters, and the per-datacenter capacity constraint (3b) couples all d_{ij} across front-end servers.

Fortunately, the above challenge can be addressed by introducing a set of auxiliary variables $a_{ij} = d_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}$, and reformulating GLB into the following problem, GLB*:

$$\begin{aligned} \text{GLB}^* \quad \min \quad & \sum_{j\in\mathcal{D}} C_j\Big(\sum_{i\in\mathcal{S}} a_{ij}\Big), \\ \text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \sum_{j\in\mathcal{D}} d_{ij}L_{ij}/D_i \le W_i, \forall i \in \mathcal{S}, \\ & \sum_{i\in\mathcal{S}} a_{ij} \le S_j, a_{ij} = d_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}, \\ & d_{ij} \ge 0, a_{ij} \ge 0, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}. \end{aligned}$$

In the new problem GLB*, auxiliary variables a_{ij} determine the objective functions with only the datacenter capacity constraints, which couple a_{ij} across i. The original variables d_{ij} control the workload conservation constraints and the performance constraints, which couple d_{ij} across j. This is the key idea that makes the minimization of the augmented Lagrangian decomposable.

The augmented Lagrangian $\mathcal{L}_\rho$ of GLB* can be readily obtained from (4). By omitting the irrelevant terms, we find that at each iteration $t + 1$, the d-minimization step is *decomposable* over i into M per-front-end server subproblems, while the a-minimization step is decomposable into N per-datacenter sub-problems. The reformulated problem GLB* can be efficiently computed in a *fully distributed* manner: each datacenter and front-end server first solves a rather small-scale subproblem; then, the front-end servers and datacenters exchange messages with each other, such local computation iterates towards convergence to the global optimum of GLB.

Our proposed distributed ADMM algorithm is shown in Algorithm 1, whose output is the optimal geographical load balancing d_{ij}, *i.e.*, the amount of workload routed from front-end server i to datacenter j. Then, based on d_{ij} and the equations listed in Table II, each datacenter j computes the optimal solution of green capacity panning, *i.e.*, the amount of power drawn from the grid, sold back to the grid, and generated by the on-site fuel cells. Note that both the per-front-end server subproblem (5) and the per-datacenter subproblem (6) are of a much smaller scale than GLB*, with only N and M variables, respectively. The objectives of the problems described above are strongly convex, so problem (5) and problem (6) can be efficiently solved with standard convex optimization techniques such as the interior-point method.

Implementation issues: The computation of the distributed ADMM algorithm can be undertaken by each local facility. However, such a fully localized implementation would make the broadcast operations at each iteration travel across the WAN that interconnects the front-end servers and datacenters, incurring heavy usage of the expensive WAN bandwidth and prolonged communication time of the broadcast operations. To address this issue, we can put the computations into a designated datacenter that has abundant server resources, and split those subproblems to the numerous servers, on which the subproblems can be solved in a parallel manner. Thanks to the high-speed intra-datacenter network that has abundant bandwidth, the bandwidth cost and communication time of the broadcast operations can be greatly reduced.

V. Solving the Multi-leader Single-follower Game

Solving the smart grid cost minimization problem SG$_j$ directly is computationally prohibitive, as e_j and n_j in SG are implicitly determined by the datacenter cost minimization problem GLB-GCP. Furthermore, the output of GLB-GCP is the input of SG$_j$. Thus, to solve SG$_j$, each smart grid needs to anticipate the response of the cloud. To address the above challenge, we model such two-stage multi-leader single-follower

Algorithm 1 Distributed Geographical Load Balancing

1: Each datacenter $j \in \mathcal{D}$ initializes a_{ij} and y_{ij} to 0, $\forall i \in \mathcal{S}$. Each front-end server $i \in \mathcal{S}$ initializes d_{ij} to 0, $\forall j \in \mathcal{D}$.

2: **d-minimization:** Each front-end server i solves the following sub-problem for d_{ij}^{t+1}, and broadcasts it to all the datacenters:

$$\begin{aligned} \min \quad & \sum_{j\in\mathcal{D}} \Big(y_{ij}^t d_{ij} + \frac{\rho}{2}(d_{ij}^2 - 2a_{ij}^t d_{ij}) \Big), \qquad (5) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{D}} d_{ij} = D_i, \sum_{j\in\mathcal{D}} d_{ij} L_{ij}/D_i \le W_i, d_{ij} \ge 0. \end{aligned}$$

3: **a-minimization:** Each datacenter j solves the following sub-problem for a_{ij}^{t+1}, and broadcasts it to all the front-end servers:

$$\begin{aligned} \min \quad & C_j\Big(\sum_{i\in\mathcal{S}} a_{ij}\Big) + \sum_{i\in\mathcal{S}} \Big(\frac{\rho}{2}(a_{ij}^2 - 2a_{ij}d_{ij}^{t+1}) - y_{ij}^t a_{ij} \Big), \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}} a_{ij} \le S_j, a_{ij} \ge 0. \qquad (6) \end{aligned}$$

4: **Dual update:** Each datacenter j updates y_{ij} for the equality constraint $d_{ij} = a_{ij}$, and broadcasts y_{ij}^{t+1} to all the front-end servers:

$$y_{ij}^{t+1} = y_{ij}^t + \rho(d_{ij}^{t+1} - a_{ij}^{t+1}).$$

5: Return to step 2 until convergence.

game as an *Equilibrium Problem with Equilibrium Constraints* (EPEC) [15], and aim to efficiently find an equilibrium point where no player can improve his/her objective by changing his/her strategy unilaterally.

For each smart grid corresponding to datacenter j, it solves the following bilevel problem BL whose upper-level is the problem SG_j, and the lower-level is the problem GLB-GCP.

$$\begin{aligned} \mathsf{BL}: \quad \min \quad & \beta_j^{\mathrm{e}}(e_j - E_j)^2 + \beta_j^{\mathrm{n}}(n_j - N_j)^2 \\ & + e_j(P_j^{\mathrm{e}} - p_j^{\mathrm{e}}) + n_j(p_j^{\mathrm{n}} - P_j^{\mathrm{n}}), \\ \text{s.t.} \quad & P_j^{\mathrm{e}\text{-}} \le p_j^{\mathrm{e}} \le P_j^{\mathrm{e}+}, P_j^{\mathrm{n}\text{-}} \le p_j^{\mathrm{n}} \le P_j^{\mathrm{n}+}, \\ & (\vec{e}, \vec{n}) \text{ solves problem GLB-GCP}. \end{aligned}$$

A. MPEC Formulation

Given the linearity and thus convexity of the lower-level problem GLB-GCP, we can replace it with its first-order optimality conditions. The optimality conditions of a linear optimization problem can be formulated through two alternative approaches: (1) Karush-Kuhn-Tucker (KKT) conditions [15]; and (2) Primal-dual transformation [15], *i.e.*, enforcing primal constraints, dual constraints and the strong duality equality.

Since the complementarity conditions derived from KKT conditions are *non-convex*, we adopt the primal-dual approach to transform the original bilevel problem BL to a single-level problem. As illustrated in Fig. 3, we replace the lower-level problem GLB-GCP with the corresponding primal constraints, dual constraints and strong duality conditions, and obtain a mathematical program with equilibrium constraints (MPEC) for each smart grid. The transformed problem MPEC_k for the smart grid corresponding to datacenter k is given in Appendix A.

Note that constraints (13c)–(13h) correspond to the primal constraints of the lower-level problem GLB-GCP, while constraints (13i)–(13m) are the dual constraints of the lower-level problem GLB-GCP. Finally, the strong duality theorem that

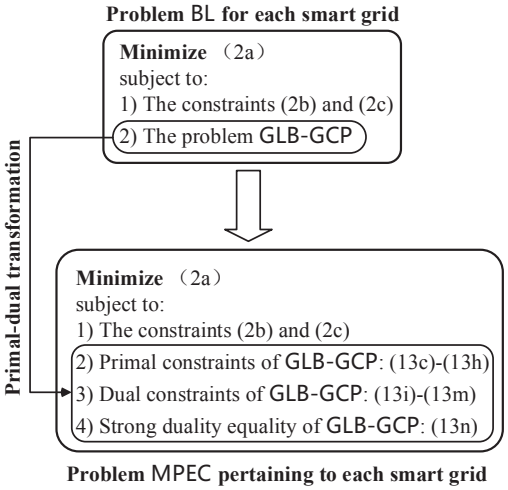

Fig. 3: From bilevel problem to single-level MPEC, via primal-dual transformation.

enforces the equality of the primal and dual objective function is denoted by constraint (13n).

The dual variables of the problem MPEC_k are indicated at their corresponding constraints in (13a)–(13n), following a colon. The dual variables associated to the shared constraints (13c)–(13n) are considered specific to each smart grid, *i.e.*, they include the subscript k.

B. EPEC Formulation

The joint consideration of the MPECs across all smart grids constitutes an Equilibrium Problem with Equilibrium Constraints (EPEC). The solution to the EPEC identifies the equilibrium of the multi-leader single-follower game. To obtain such a solution, the optimality conditions of the smart grid MPECs are to be first derived.

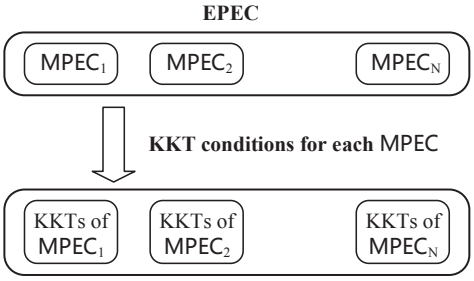

Fig. 4: EPEC and its optimality conditions.

In Sec. V-A, when replacing the lower-level problem GLB-GCP with its optimality conditions, we have avoided the non-convex and complicated complementarity conditions by relying on primal-dual transformation. However, for the problem MPEC_k, we can see that it is nonlinear, thus the application of the primal-dual transformation is not straightforward here. Hence we resort to the KKT conditions to replace each problem MPEC_k with its optimality conditions for each smart grid, and to attain the optimality conditions of the EPEC, as shown in Fig. 4.

The optimality conditions associated with EPEC include the following three types.

1) Primal equality constraints of the problem MPEC_k for each smart grid.

2) Equality constraints generated by differentiating the corresponding Lagrangian associated to each problem MPEC_k.
3) Complementarity conditions related to the inequality constraints of problem MPEC_k.

Specifically, the primal equality constraints of the problem MPEC_k consist of:

$$(13c), (13f), (13i) - (13l), (13n), \forall k \in \mathcal{D}. \tag{7}$$

By differentiating the Lagrangian function L_k of the problem MPEC_k, we further obtain the equality constraints (14) corresponding to the second type of conditions (Appendix B).

The complementarity conditions included in the KKT conditions associated to the EPEC are given by (15) in Appendix C. Here the complementarity condition $x \geq 0, y \geq 0$ and $xy = 0$ is denoted in the form of $0 \leq x \perp y \geq 0$.

Remark: The products of decision variables and complementarity conditions make the optimality conditions (7), (14), (15) nonlinear and non-convex, and thus highly challenging. Though those non-convexity can be tackled by existing methods [20] such as PATH and Nonlinear Programming Reformulation with small amount of computation, a critical disadvantage of these methods is that even they terminate properly, they can only guarantee to find a local equilibrium. On the other hand, smart grids generally deploy sufficient computing capacity to run their business, and computation burden is not a primary concern. Thus, towards accurate and efficient solutions, we propose to *exactly* linearize the optimality conditions of EPEC, and derive the resulting conditions that are mixed-integer and linear.

C. Reformulating as a Linear Problem Without Approximation

The optimality conditions (7),(14),(15) contain the following non-linearities:

1) The complementarity conditions (15).
2) The products of decision variables ($e_j p_j^{\mathrm{e}}$ and $n_j p_j^{\mathrm{n}}$) in the strong duality equalities (13m) included in (7).
3) The products of decision variables in (14a), (14b), (14d), (14e), (14f) and (14g). The common variables of such non-linear terms are the dual variables ψ_k that correspond to the strong duality equalities (13m).

Linearizing the complementarity conditions (15): Each complementarity condition in the form $0 \leq x \perp y \geq 0$ can be replace by the *equivalent* disjunctive formulation

$$x \geq 0, y \geq 0, x \leq \theta M^x, y \leq (1-\theta) M^y, \theta \in \{0,1\}. \tag{8}$$

where M^x and M^y are large enough positive constants.

Linearizing the strong duality equalities (13n): As explained in Sec. V-A, the strong duality equality from the primal-dual transformation is equivalent to the set of complementarity conditions obtained through KKT conditions. For linearity, we replace the strong duality equalities (13n) with their equivalent complementarity conditions below related to the inequality constraints. These complementarity conditions can be linearized through (8).

$$0 \leq (S_j - \sum_{i \in \mathcal{S}} d_{ij}) \perp \nu_j^{\mathrm{S}} \geq 0, \forall j, \tag{9a}$$

$$0 \leq (W_i - \sum_{j \in \mathcal{D}} d_{ij} L_{ij}/D_i) \perp \nu_i^{\mathrm{W}} \geq 0, \forall i, \tag{9b}$$

$$0 \leq e_j \perp \nu_j^{\mathrm{e}} \geq 0, 0 \leq d_{ij} \perp \nu_{ij}^{\mathrm{d}} \geq 0, \forall i, j, \tag{9c}$$

$$0 \leq g_j \perp \nu_j^{\mathrm{g^-}} \geq 0, 0 \leq (G_j - g_j) \perp \nu_j^{\mathrm{g^+}} \geq 0, \forall j, \tag{9d}$$

$$0 \leq n_j \perp \nu_j^{\mathrm{n^-}} \geq 0, 0 \leq (R_j - n_j) \perp \nu_j^{\mathrm{n^+}} \geq 0, \forall j, \tag{9e}$$

Using complementarity conditions to replace strong duality equality (13n) may seem against the statement made in Sec. V-A, *i.e.*, the primal-dual formulation is more computationally efficient than the KKT conditions. However, such efficiency is embodied when deriving (14)–(15) from the MPEC, since the strong stationary conditions (7), (14), (15) would have been much more complicated if we replace the lower-level problem $\mathsf{GLB\text{-}GCP}$ with its KKT conditions.

Linearizing the nonlinear terms involving ψ_k: We propose to parameterize the problem in the variables ψ_k, since these are dual variables associated to problem MPEC, and the set of dual variables at any solution forms a ray and has some degrees of freedom [18]. Moreover, as there is only one ψ_k for each smart grid, the parameterizations can be easily implemented to characterize the feasible region of the problem.

Hence, the nonlinear terms in (14a), (14b), (14d), (14e), (14f) and (14g) become linear if parameterizing problem (7), (14), (15) in dual variables ψ_k. As for the selection of the value of the dual variables ψ_k, note that the combination of the constraints requires that the dual variables ψ_k to be non-negative.

In the above way, we transform the nonlinear EPEC into a system of linear equations and inequalities that involve continuous and binary variables, which we refer to as the linear version of EPEC. **Since the above linearization is performed without approximation, the linear EPEC is equivalent to the nonlinear EPEC, and thus the original multi-leader single-follower game.** However, the transformed linear system generally has multiple solutions, and choosing a meaningful equilibrium for the game may be challenging.

D. Classification of Equilibrium

In order to find a particular meaningful equilibrium among all the possible equilibriums of the EPEC, we define a linear classification function which serves as the objective function of the new MILP whose constraints consist of the linear version of the EPEC. Note that this additional objective function is not the objective function of either the lower or the upper level. The proposed MILP structure is:

$$\begin{array}{ll} \min & \text{Linear classification function,} \\ \text{s.t.} & \text{MILP version of the conditions (7), (14), (15).} \end{array}$$

A wide range of classification functions are possible in the cloud-smart grid interaction scenario. For example: Total cost of all the smart grids; Aggregated energy cost of the geo-distributed cloud; or Social cost of all the smart grids and the geo-distributed cloud.

However, all the three objective functions above contain non-linear terms. For example, the aggregated energy cost of the geo-distributed cloud contains the products of decision variables (*e.g.*, $p_j^{\mathrm{e}} e_j$), and the cost of each smart grid contains both quadric terms and products of decision variables (*e.g.*, $(e_j - E_j)^2$ and $p_j^{\mathrm{e}} e_j$, respectively). Fortunately, both the quadric terms and products of decision variables can be well-approximated using linear objective functions. For example,

the product of two decision variables can be linearized by discretizing one of the decision variables, while the quadratic term can be approached by a piecewise linear function.

As a case study, we describe in detail how to linearize the total cost of all the smart grids, and then yield the linear approximation of the classification function.

Discretize grid power draw. We now deal with the products of decision variables, like $p_j^{\mathrm{e}} e_j$. The basic idea is to approximate the continuous variable e_j by a set of discrete values $\{e_{jm}, m = 0, 1, ..., U\}$, where $U = 2^{L_1}$ for some non-negative integer L_1. For example, given that the variables e_j are in the range $[0, \alpha_j S_j]$, the discrete approximation is written as $e_j = \Delta_j^1 \sum_{l_1=0}^{L_1} 2^l q_{jm}$, where $\Delta_j^1 = \alpha_j S_j / U$ is the stepsize, and q_{jm} is a binary variable. The above expression is called a binary expansion [30], multiplying both sides by p_j^{e}, and replacing $q_{jm} p_j^{\mathrm{e}}$ with a new variable w_{jm}^1, we obtain:

$$p_j^{\mathrm{e}} e_j = \Delta_j^1 \sum_{l=0}^{L_1} 2^l w_{jm}^1. \tag{10}$$

This expression allows us to replace the product $p_j^{\mathrm{e}} e_j$ with the linear expression on the right-hand side. In turn, the product of variables in $w_{jm}^1 = q_{jm} p_j^{\mathrm{e}}$ is transformed to the IF-THEN relation: if $q_{jm} = 0$, then $w_{jm}^1 = 0$; if $q_{jm} = 1$, then $w_{jm}^1 = p_j^{\mathrm{e}}$. This relation can be modeled as:

$$0 \le w_{jm}^1 \le M^e q_{jm}, 0 \le p_j^{\mathrm{e}} - w_{jm}^1 \le M^e (1 - q_{jm}). \tag{11}$$

Where M^e is a scalar value that is large enough. Similarly, we can also approximate $p_j^{\mathrm{n}} n_j$ by $\Delta_j^2 \sum_{l_2=0}^{L_2} 2^{l_2} w_{jm}^2$.

Piecewise linear function. We formulate a piecewise linear function to approximate the quadratic terms. Given $n_j \in [0, R_j]$, at a set of points $\{0, \frac{R_j}{V}, 2\frac{R_j}{V}, ..., R_j\}$ the curve $(n_j - N_j)^2$ is approximated by the tangent lines. Let us refer to the x-axis value of the intersection points of these lines as IP_{js}, where $s \in \{0, 1, 2, ..., V\}$ is the index of intersection points (including the start point $IP_{j0} = 0$ of the first tangent line and the end point $IP_{jV} = R_j$ of the last tangent line) and LS_{js} represents the slope of tangent lines in a total number of V. We use LD_{js} to represent the part of e_j between the intersection points $s-1$ and s. The binary variable b_{js} is one if the variable LD_{js} is at its upper bound $IP_{js} - IP_{js-1}$, then we define $\bar{n}_j$ to linearly approximate the quadratic function $(n_j - N_j)^2$ as follows:

$$(IP_{js} - IP_{js-1}) b_{js} \le LD_{js}, \forall j, s = 1, 2, ..., V, \tag{12a}$$
$$(IP_{js} - IP_{js-1}) b_{js} \ge LD_{js}, \forall j, s = 1, 2, ..., V, \tag{12b}$$
$$n_j = \sum_{s=1}^{V} LD_{js}, \forall j, \tag{12c}$$
$$\bar{n}_j = E_j^2 + \sum_{s=1}^{V} LD_{js} LS_{js}, \forall j, \tag{12d}$$
$$LS_{js} = 2(\frac{R_j}{V} s - E_j), \forall j, s, \tag{12e}$$
$$IP_{j0} = 0, IP_{jV} = R_j, \forall j; \tag{12f}$$
$$IP_{js} = \frac{R_j}{2V}(2s - 1), \forall j, s = 1, 2, ..., V - 1. \tag{12g}$$

Similarly, we can also approximate $(e_j - E_j)$ with $\bar{e}_j$. Finally, applying the above results to the cost function of each smart grid, yielding a linear classic function that represents the approximation of the total cost of all the smart grids. And the MILP that aims to find a meaningful equilibrium can be summarized as:

$$\min \quad \sum_{j \in \mathcal{D}} \Big\{ \beta_1 \bar{e}_j + \beta_2 \bar{n}_j - \Delta_j^1 \sum_{l_1=0}^{L_1} 2^{l_1} w_{jm}^1 + \Delta_j^2 \sum_{l_2=0}^{L_2} 2^{l_2} w_{jm}^2 \Big\},$$

s.t. (1): Linearize products decision variables, of the form (10)–(11),

(2): Linearize quadratic function, of the form (12),

(3): Linear version of the EPEC.

Though the above MILP is of larger scale than GLB-GCP, it can be efficiently solved using conventional branch-and-cut techniques in a **parallel** manner [24].

VI. Performance Evaluation

We have conducted trace-driven simulations to evaluate the practical economic benefits of the proposed incentive mechanism. The simulations are based on real-world datacenter workload traces and IEEE 14-bus test systems with real-world generation and demand data.

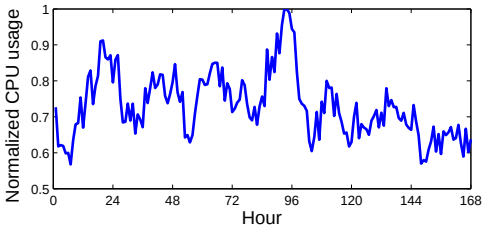

Fig. 5: Normalized CPU usage trace from a Google cluster.

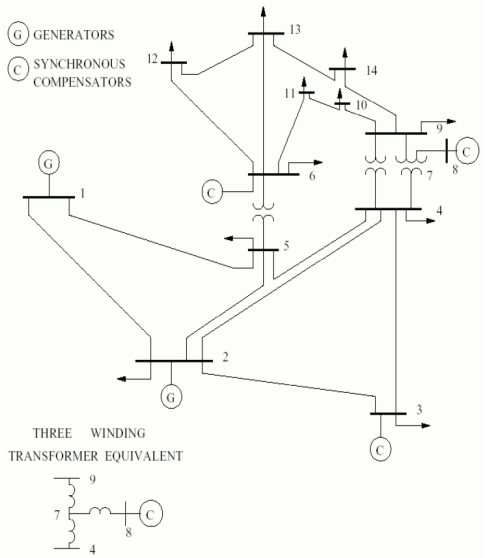

Fig. 6: Topology of the IEEE 14-bus test system.

A. Simulation Setup

Geo-distributed cloud: we consider six Google datacenters in the U.S. as a representative geo-distributed cloud. Following a recent report on the number of servers owned by Google, each datacenter's capacity is set to 2×10^5 processing servers. We use the CPU usage extracted from Google cluster-usage data [5], as shown in Fig. 5 to represent the request traffic of an interactive cloud service. The workload exhibits great variability and a clear diurnal pattern, typical for interactive cloud service. To imitate the geographical distribution of

requests, we split this total workload among the $M = 10$ front-end servers following a normal distribution [31]. Each round-trip time L_{ij} is calculated according to the aforementioned empirical approximation $L_{ij} = t_{ij} \times 0.02\text{ms/Km}$, where the geographical distance t_{ij} is obtained from Google Maps. We set the energy usage efficiency parameter $\alpha_j = 250\text{W}$ for each datacenter j, *i.e.*, each unit amount of request that demands one server leads to 250W power consumption of the datacenter. We also enforce the average perceived latency at each front-end server i to be within $W_i = 50\text{ms}$.

Smart grids: we use the IEEE 14-bus test system [11] illustrated in Fig. 6 to represent a smart grid that serves a datacenter. The arrows in Fig. 6 represent various power loads such as datacenter power demand, the synchronous condensers at buses 3, 6 and 8 can be replaced by power loads or renewable generators, and two other generators are connected to buses 1 and 2. To distinguish among the six smart grids, we place the datacenters at different buses of the test system, and use different configurations of renewable generations and power demands. The demand profiles and renewable generations are taken form the SCE load profile [21] and the NREL datasets [10], respectively. The desired levels of datacenter power consumption and net metering that minimize the voltage violation frequency, E_j and N_j, respectively, are computed by using MatPower [21]. Unit electricity cost at each smart grid is set to 0.6X the 2011 annual average day-ahead on peak price at the corresponding local markets [31]. The original net metering price is set to the unit electricity cost. The location of each datacenter and the corresponding unit electricity cost and original net metering price are listed in Table III.

TABLE III: Unit electricity cost and original net metering price ($USD/MWh) at different datacenter locations.

Council Bluffs, IA	42.73	Berkeley County, SC	44.44
The Dalles, OR	32.57	Lenoir, NC	40.68
Mayes County, OK	36.41	Douglas County, GA	39.97

B. Performance Evaluation

For comparison, we further implement and evaluate a benchmark approach, in which the smart grids only update the electricity price via the multi-leader single-follower game in a realtime manner as in [28], [29], [26], while the net metering price is fixed at some percentage of the electricity cost. Hereinafter, we call the benchmark Unilateral and our solution Bilateral.

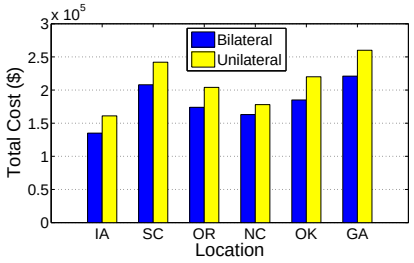

Fig. 7: Total cost of each smart grid under different pricing schemes.

Efficiency of the bilateral electricity trade pricing. We first examine the efficiency of the proposed bilateral electricity trade pricing scheme, in terms of cost reduction. Fig. 7 plots the total cost of each smart grid under different pricing schemes. It can be seen that, by pricing the two-way electricity flow in a realtime manner, the cost of all the smart grids can be effectively reduced. We further demonstrate the efficiency of the proposed pricing scheme in improving the smart grids' stability in a more straightforward manner. We define the aggregated demand deficit (quantified by $|e_j - E_j| + |n_j - N_j|$) as the gap between the j-th datacenter's actual capacity profile (e_j, n_j) and the most stable capacity profile (E_j, N_j) to capture the stability of the smart grids. Fig. 8 shows that the stability of each smart grid can be largely improved when realtime pricing for bilateral electricity trade is introduced.

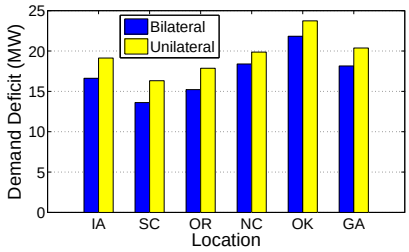

Fig. 8: Time-averaged aggregated demand deficit for each smart grid under different schemes.

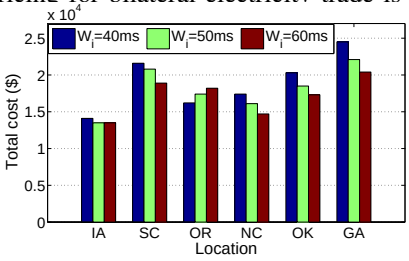

Fig. 9: Total cost for each smart grid under different performance target W_i.

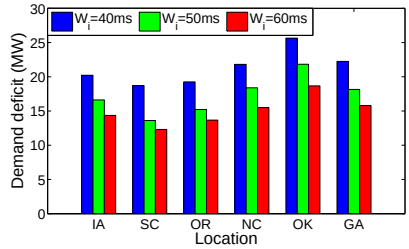

Fig. 10: Time-averaged aggregated demand deficit of each smart grid under different performance target W_i.

Impact of the performance target of the cloud service on smart grids. Fig. 9 compares the total cost of each smart grid under different performance targets (*i.e.,* the W_i discussed in Sec. III-C) of the cloud service. We have the following observations: (1) Except for the smart grid at OR, the total cost of each of the other five smart grids declines as the performance target W_i increases. (2) However, for the smart grid at OR, the total cost ascends with the increase of W_i. The intuition behind this exception is that the increase of W_i incurs excessive workload being distributed to the datacenter located at OR with the lowest unit electricity cost, and exacerbates the stability of the smart grid at OR. We further compare the time-averaged aggregated demand deficit for each smart grid under different performance target in Fig. 10, we observe that the time-averaged aggregated demand deficit of each smart grid declines as the performance target W_i increases. This is because a tighter performance target requires more workload

to be served by closer datacenters, and hence reduces the flexibility of geographical load balancing on demand response.

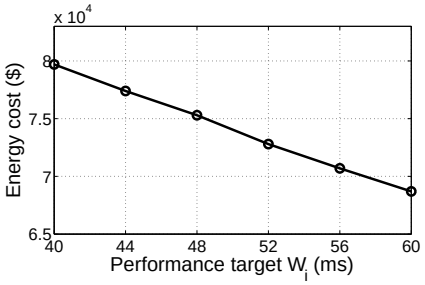

Fig. 11: Total energy cost of the cloud under different performance target W_i.

Impact of the performance target on the energy cost of the cloud. It is intuitive that if the prices of grid power and net metering do not depend on the performance target of the cloud, then the cloud can save more energy cost with a looser performance target, by making the geographical load balancing price-aware. However, when the price decisions consider the performance target, as the game presented in this work, does the above intuition still hold? To answer this question, we plot the total energy cost of the cloud under various performance targets in Fig. 11. We can see that as the energy cost of the cloud reduces as its performance target W_i increases. This is easy to interpret from the view of economic, *i.e.,* the more budget the customers have, they would benefit more from the full competition among many producers.

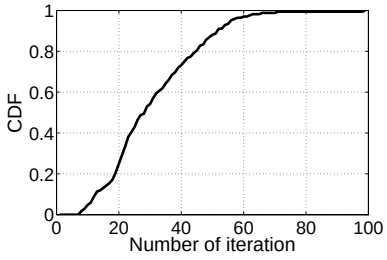

Fig. 12: CDF of the number of iterations till convergence.

Convergence of the distributed geographical load balancing. We now examine the convergence of our ADMM based distributed geographical load balancing algorithm. Fig. 12 plots the CDF of the number of iterations our algorithm takes to achieve convergence for the 168 runs. The algorithm is able to converge within 50 iterations for 90% of the total runs. Furthermore, the fastest run takes only 8 iterations, and our algorithm takes at most 98 iterations to converge. This demonstrates the fast convergence of our ADMM based algorithm.

Running time. Since we do not have enough server resource to experiment with a parallel implementation, we evaluate the ADMM algorithm on an Intel Xeon E5-2670 server with 8-core CPU (2.6G) and 8GB DDR3 memory, where each per-front-end server and per-datacenter sub-problem is solved sequentially. We observe that one hourly instance takes 0.13s on average, considering there are 10 front-end servers and 6 datacenters, thus if implemented in a fully parallel manner, each hourly instance takes about 0.008 seconds. For the MILP algorithm, we observe that one hourly instance takes 2.54 seconds on average. Note that in realistic cloud systems, up to tens of thousand of front-end servers may be deployed. In this case, a parallel implementation of the MILP algorithm is desired to reduce the computation time, and we leave the algorithm design amenable to parallel implementation as future work.

VII. Limitation and future work

Ramping constraints for fuel cells. For some traditional thermal generators such as steam turbine, gas turbine, and even solid oxide fuel cell (SOFC), power generation is constrained by ramping cost/time and start-up cost/time [22], due to the requirement of extremely high operating temperatures. While for proton exchange membrane fuel cell (PEMFC) that is being used to power datacenters [33], it can work in a low temperature range with rather short start-up time (as low as 1 second [3]) and fast ramping rate (*e.g.*, 192W/s for a 10 kW PEMFC system [33]), leading to very low start-up cost and ramping cost. Thus, for simplicity, as in recent study on power datacenters with fuel cells [33], the above operation constraints are not considered in our model. In the future, to make our model more general and applicable to other fuel cell systems, we would like to incorporate the above operational constraints into our model. In particular, the formulation of those complicated constraints can be obtained by adapting the formulation for traditional thermal generators [22], efficient distributed energy cost minimization for the cloud and meaningful price equilibriums for the smart grids are to be investigated.

Parallel implementation of the MILP algorithm. The MILP is solved in a centralized fashion in the evaluations. However, some realworld clouds have up to tens of thousand of front-end servers, thus the corresponding MILP would have a very large scale. Then, a parallel implementation of the MILP algorithm is expected to efficiently reduce the computation time. Though the classic branch-and-cut method for MILP can be parallelized [25] to reduce computation time, it is still difficult to do this in such as way that: (1) the amount of time to solve each of the resulting smaller MILPs is approximately equal, in case that some servers will become idle before the solution has been found; and (2) the total amount of power required to solve the subproblems should not far exceed the amount of power required to solve the original problem on a single server. We hope to address these challenges in future work.

Joint optimization on the servers and networks. In this paper, we focus on the server-related energy consumption and ignore the energy consumption of the intra-datacenter network and the wide-area network. While for applications such as video streaming, they require few server resource but incur huge amounts of traffic and power consumption at the network side. For such applications, network side energy optimization is critical and the recent work has proposed energy-efficient routing scheme [34]. However, for geo-distributed datacenters, the joint energy optimization on both the server and network sides has been rarely studied, and we hope to tackle this problem in future work.

VIII. Concluding Remarks

This work presented a realtime pricing design for demand response from a geo-distributed cloud, based on a multi-leader single-follower model. Two important considerations are embodied in the model: (i) Two-way electricity flow between smart grids and large datacenters with hybrid on-site generation capabilities. (ii) The geo-distributed nature of large cloud systems, and hence the potential competition among smart grids that serve different datacenters of the cloud. At the datacenter side, by using the technique of alternating direction of multipliers, the geographical load balancing and green capacity planning are performed in a distributed manner to minimize the energy cost of the cloud. At the smart grid side, in quest for a practical equilibrium of the game, we transform the multi-leader single-follower game into a much simpler mixed-integer linear programming problem, by employing techniques such as mathematical programming with equilibrium constraints, equilibrium problem with equilibrium constraints and exact linearization. Extensive trace-driven evaluations running on the IEEE 14-bus test system demonstrate the economical benefits of the proposed realtime pricing.

Acknowledgment

The research was supported in part by a grant from the National Natural Science Foundation of China (NSFC) under grant No.61520106005, by a grant from National 973 Basic Research Program under grant No.2014CB347800. The corresponding author is Fangming Liu.

Appendix

A. The problem MPEC

We use $\nu_i^{\mathrm{D}}, \nu_j^{\mathrm{S}}, \nu_i^{\mathrm{W}}, \nu_j, \nu_j^{\mathrm{e}}, \nu_j^{\mathrm{g\text{-}}}, \nu_j^{\mathrm{g+}}, \nu_j^{\mathrm{n\text{-}}}, \nu_j^{\mathrm{n+}}, \nu_j^{\mathrm{d}}$ to denote the Lagrangian multipliers of the constraints (1a)–(1f) in problem GLB-GCP, respectively. Then, the problem MPEC_k for each smart grid k defined in Sec. V-A is given as follows:

MPEC_k

$$\begin{aligned}
\min \quad & \beta_k(e_k - E_k)^2 + \beta_k(n_k - N_k)^2 \\
& + (P_k^{\mathrm{e}} - p_k^{\mathrm{e}})e_k + (p_k^{\mathrm{n}} - P_k^{\mathrm{n}})n_k, \\
\text{s.t.} \quad & P_k^{\mathrm{e\text{-}}} \le p_k^{\mathrm{e}} \le P_k^{\mathrm{e+}} \quad : \mu_k^{\mathrm{pe\text{-}}}, \mu_k^{\mathrm{pe+}}, && \text{(13a)} \\
& P_k^{\mathrm{n\text{-}}} \le p_k^{\mathrm{n}} \le P_k^{\mathrm{n+}} \quad : \mu_k^{\mathrm{pn\text{-}}}, \mu_k^{\mathrm{pn+}}, && \text{(13b)} \\
& \sum_{j\in\mathcal{D}} d_{ij} = D_i \quad : \mu_{ik}^{\mathrm{D}}, \forall i, && \text{(13c)} \\
& \sum_{i\in\mathcal{S}} d_{ij} \le S_j \quad : \mu_{jk}^{\mathrm{S}}, \forall j, && \text{(13d)} \\
& \sum_{j\in\mathcal{D}} d_{ij}L_{ij}/D_i \le W_i \quad : \mu_{ik}^{\mathrm{W}}, \forall i, && \text{(13e)} \\
& e_j + g_j + R_j - n_j = \alpha_j \sum_{i\in\mathcal{S}} d_{ij} \quad : \mu_{jk}, \forall j, && \text{(13f)} \\
& e_j \ge 0, 0 \le g_j \le G_j, 0 \le n_j \le R_j, \\
& \quad : \mu_{jk}^{\mathrm{e}}, \mu_{jk}^{\mathrm{g\text{-}}}, \mu_{jk}^{\mathrm{g+}}, \mu_{jk}^{\mathrm{n\text{-}}}, \mu_{jk}^{\mathrm{n+}}, \forall j, && \text{(13g)} \\
& d_{ij} \ge 0 \quad : \mu_{ijk}^{\mathrm{d}}, \forall i, j, && \text{(13h)} \\
& p_j^{\mathrm{e}} - \nu_j^{\mathrm{e}} + \nu_j = 0 \quad : \phi_{jk}^{\mathrm{e}}, \forall j && \text{(13i)} \\
& p_g + \nu_j - \nu_j^{\mathrm{g\text{-}}} + \nu_j^{\mathrm{g+}} = 0 \quad : \phi_{jk}^{\mathrm{g}}, \forall j, && \text{(13j)} \\
& -p_j^{\mathrm{n}} - \nu_j - \nu_j^{\mathrm{n\text{-}}} + \nu_j^{\mathrm{n+}} = 0 \quad : \phi_{jk}^{\mathrm{n}}, \forall j, && \text{(13k)} \\
& \nu_i^{\mathrm{D}} + \nu_j^{\mathrm{S}} + \frac{L_{ij}}{D_i}\nu_i^{\mathrm{W}} - \alpha_j\nu_j - \nu_{ij}^{\mathrm{d}} = 0 \quad : \phi_{ijk}^{\mathrm{d}}, \forall i, j, && \text{(13l)} \\
& \nu_j^{\mathrm{S}}, \nu_i^{\mathrm{W}}, \nu_j^{\mathrm{e}}, \nu_j^{\mathrm{g\text{-}}}, \nu_j^{\mathrm{g+}}, \nu_j^{\mathrm{n\text{-}}}, \nu_j^{\mathrm{n+}}, \nu_{ij}^{\mathrm{d}} \ge 0, \\
& \quad : \varphi_{jk}^{\mathrm{S}}, \varphi_{ik}^{\mathrm{W}}, \varphi_{jk}^{\mathrm{e}}, \varphi_{jk}^{\mathrm{g\text{-}}}, \varphi_{jk}^{\mathrm{g+}}, \varphi_{jk}^{\mathrm{n\text{-}}}, \varphi_{jk}^{\mathrm{n+}}, \varphi_{ijk}^{\mathrm{d}}, \forall i, j, && \text{(13m)} \\
& \sum_{j\in\mathcal{D}} \left\{ e_j p_j^{\mathrm{e}} + g_j p_g - n_j p_j^{\mathrm{n}} \right\} + \sum_{i\in\mathcal{S}} \nu_i^{\mathrm{W}} W_i \\
& + \sum_{j\in\mathcal{D}} \left\{ \nu_j^{\mathrm{S}} S_j + \nu_j^{\mathrm{g+}} G_j + \nu_j^{\mathrm{n+}} R_j \right\} = 0 \quad : \psi_k. && \text{(13n)}
\end{aligned}$$

B. Differentiating the Lagrangian

By differentiating the Lagrangian function L_k of the problem MPEC, we obtain the equality constraints (14):

$$\frac{\partial L_k}{\partial p_k^{\mathrm{e}}} = -e_k - \mu_k^{\mathrm{pe\text{-}}} + \mu_k^{\mathrm{pe+}} + \phi_{jk}^{\mathrm{e}} + \psi_k e_k = 0, \forall k, j = k \quad \text{(14a)}$$

$$\frac{\partial L_k}{\partial p_k^{\mathrm{n}}} = n_k - \mu_k^{\mathrm{pn\text{-}}} + \mu_k^{\mathrm{pn+}} - \phi_{jk}^{\mathrm{n}} - \psi_k n_k = 0, \forall k, j = k \quad \text{(14b)}$$

$$\frac{\partial L_k}{\partial d_{ij}} = \mu_{ik}^{\mathrm{D}} + \mu_{jk}^{\mathrm{S}} + \frac{L_{ij}}{D_i}\mu_{ik}^{\mathrm{W}} + \alpha_j\mu_{jk} - \mu_{ijk}^{\mathrm{d}} = 0, \forall i, j, k, \quad \text{(14c)}$$

$$\frac{\partial L_k}{\partial e_j} = \mu_{jk} - \mu_{jk}^{\mathrm{e}} + \psi_k p_j^{\mathrm{e}} = 0, \forall j \ne k, k, \quad \text{(14d)}$$

$$\frac{\partial L_k}{\partial e_j} = 2\beta_k^{\mathrm{e}}(e_k - E_k) + P_k^{\mathrm{e}} - p_k^{\mathrm{e}} + \mu_{jk} - \mu_{jk}^{\mathrm{e}} + \psi_k p_j^{\mathrm{e}} = 0, \forall j = k, k, \quad \text{(14e)}$$

$$\frac{\partial L_k}{\partial n_j} = -\mu_{jk} - \mu_{jk}^{\mathrm{n\text{-}}} + \mu_{jk}^{\mathrm{n+}} - \psi_k p_j^{\mathrm{n}} = 0, \forall j \ne k, k, \quad \text{(14f)}$$

$$\frac{\partial L_k}{\partial n_j} = 2\beta_k^{\mathrm{n}}(n_k - N_k) + p_k^{\mathrm{n}} - P_k^{\mathrm{n}} - \mu_{jk} - \mu_{jk}^{\mathrm{n\text{-}}} + \mu_{jk}^{\mathrm{n+}} - \psi_k p_j^{\mathrm{n}} = 0, \quad \forall j = k, k, \quad \text{(14g)}$$

$$\frac{\partial L_k}{\nu_i^{\mathrm{D}}} = \sum_j \phi_{ijk}^{\mathrm{d}} = 0, \frac{\partial L_k}{\nu_i^{\mathrm{W}}} = \sum_j \frac{L_{ij}}{D_i}\phi_{ijk}^{\mathrm{d}} - \varphi_{ik}^{\mathrm{W}} = 0, \forall i, k, \quad \text{(14h)}$$

$$\frac{\partial L_k}{\partial g_j} = \mu_{jk} - \mu_{jk}^{\mathrm{g\text{-}}} + \mu_{jk}^{\mathrm{g+}} + \psi_k p_g = 0, \forall j, k, \quad \text{(14i)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{S}}} = \sum_i \phi_{ijk}^{\mathrm{d}} - \varphi_{jk}^{\mathrm{S}} + \psi_k S_j = 0, \forall j, k, \quad \text{(14j)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{e}}} = -\phi_{jk}^{\mathrm{e}} - \varphi_{jk}^{\mathrm{e}} = 0, \forall j, k, \quad \text{(14k)}$$

$$\frac{\partial L_k}{\nu_{ij}^{\mathrm{d}}} = -\phi_{ijk}^{\mathrm{d}} - \varphi_{ijk}^{\mathrm{d}} = 0, \forall i, j, k, \quad \text{(14l)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{g\text{-}}}} = -\phi_{jk}^{\mathrm{g}} - \varphi_{jk}^{\mathrm{g\text{-}}} = 0, \forall j, k, \quad \text{(14m)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{g+}}} = \phi_{jk}^{\mathrm{g}} - \varphi_{jk}^{\mathrm{g+}} + \psi_k G_j = 0, \forall j, k, \quad \text{(14n)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{n\text{-}}}} = -\phi_{jk}^{\mathrm{n}} - \varphi_{jk}^{\mathrm{n\text{-}}} = 0, \forall j, k, \quad \text{(14o)}$$

$$\frac{\partial L_k}{\nu_j^{\mathrm{n+}}} = \phi_{jk}^{\mathrm{n}} - \varphi_{jk}^{\mathrm{n+}} + \psi_k R_j = 0, \forall j, k. \quad \text{(14p)}$$

C. Complementarity Conditions

The complementarity conditions included in the KKT conditions associated to the EPEC are given by (15):

$$0 \le (p_k^{\mathrm{e}} - P_k^{\mathrm{e\text{-}}}) \perp \mu_k^{\mathrm{pe\text{-}}} \ge 0, 0 \le (P_k^{\mathrm{e+}} - p_k^{\mathrm{e}}) \perp \mu_k^{\mathrm{pe+}} \ge 0, \forall k, \quad \text{(15a)}$$

$$0 \le (p_k^{\mathrm{n}} - P_k^{\mathrm{n\text{-}}}) \perp \mu_k^{\mathrm{pn\text{-}}} \ge 0, 0 \le (P_k^{\mathrm{n+}} - p_k^{\mathrm{n}}) \perp \mu_k^{\mathrm{pn+}} \ge 0, \forall k, \quad \text{(15b)}$$

$$0 \le (S_j - \sum_{i\in\mathcal{S}} d_{ij}) \perp \mu_{jk}^{\mathrm{S}} \ge 0, 0 \le g_j \perp \mu_{jk}^{\mathrm{g\text{-}}} \ge 0, \forall j, k, \quad \text{(15c)}$$

$$0 \le (W_i - \sum_{j\in\mathcal{D}} d_{ij}L_{ij}/D_i) \perp \mu_{ik}^{\mathrm{W}} \ge 0, 0 \le \nu_i^{\mathrm{W}} \perp \varphi_{ik}^{\mathrm{W}} \ge 0, \forall i, k, \quad \text{(15d)}$$

$$0 \le (G_j - g_j) \perp \mu_{jk}^{\mathrm{g+}} \ge 0, 0 \le \nu_j^{\mathrm{S}} \perp \varphi_{jk}^{\mathrm{S}} \ge 0, \forall j, k, \quad \text{(15e)}$$

$$0 \le d_{ij} \perp \mu_{ijk}^{\mathrm{d}} \ge 0, 0 \le \nu_{ij}^{\mathrm{d}} \perp \varphi_{ijk}^{\mathrm{d}} \ge 0, \forall i, j, k, \quad \text{(15f)}$$

$$0 \le e_j \perp \mu_{jk}^{\mathrm{e}} \ge 0, 0 \le n_j \perp \mu_{jk}^{\mathrm{n\text{-}}} \ge 0, \forall j, k, \quad \text{(15g)}$$

$$0 \le (R_j - n_j) \perp \mu_{jk}^{\mathrm{n+}} \ge 0, 0 \le \nu_j^{\mathrm{n\text{-}}} \perp \varphi_{jk}^{\mathrm{n\text{-}}} \ge 0, \forall j, k, \quad \text{(15h)}$$

$$0 \leq \nu_j^{\text{n+}} \perp \varphi_{jk}^{\text{n+}} \geq 0, 0 \leq \nu_j^{\text{e}} \perp \varphi_{jk}^{\text{e}} \geq 0, \forall j, k, \quad (15i)$$

$$0 \leq \nu_j^{\text{g-}} \perp \varphi_{jk}^{\text{g-}} \geq 0, 0 \leq \nu_j^{\text{g+}} \perp \varphi_{jk}^{\text{g+}} \geq 0, \forall j, k. \quad (15j)$$

REFERENCES

[1] Environmental Responsibility Report. https://www.apple.com/environment/reports/.

[2] Facebook's Altoona, Iowa Data Center to be Completely Wind-powered. http://www.slashgear.com/facebooks-altoona-iowa-data-center-to-be-completely-wind-powered-13305335/.

[3] Fuel cell. https://en.wikipedia.org/wiki/Fuel-cell.

[4] German power prices negative over weekend. http://energytransition.de/2014/05/german-power-prices-negative-over-weekend.

[5] Google Cluster Data. https://code.google.com/p/googleclusterdata/.

[6] Google Green. http://www.google.com/green/bigpicture.

[7] Inside one of the world's largest data centers. http://www.cnet.com/news/inside-one-of-the-worlds-largest-data-centers/.

[8] Microsoft Environment. http://www.microsoft.com/environment/our-commitment/our-footprint.aspx.

[9] Net Metering. http://en.wikipedia.org/wiki/Net-metering.

[10] NREL Data and Resources. http://www.nrel.gov/electricity/transmission/data-resources.html.

[11] Power Systems Test Case Archive. http://www.ee.washington.edu/research/pstca/.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[13] T. Chen, Y. Zhang, X. Wang, and G. B. Giannakis. Robust workload and energy management for sustainable data centers. *IEEE Journal on Selected Areas in Communications*, 34(3):651 – 664, 2016.

[14] W. Deng, F. Liu, H. Jin, B. Li, and D. Li. Harnessing renewable energy in cloud datacenters: Opportunities and challenges. *IEEE Network Magzine*, 28(1):48 – 55, 2014.

[15] S. A. Gabriel, A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz. Complementarity modeling in energy markets. *International Series in Operations Research and Management Science*, 180:1–629, 2012.

[16] P. X. Gao, A. R. Curtis, B. Wang, and S. Keshav. It's Not Easy Being Green. In *Proc. of ACM SIGCOMM*, 2012.

[17] A. M. Ibrahim, A. A. Zewail, and A. Yener. Green distributed storage using energy harvesting nodes. *IEEE Journal on Selected Areas in Communications*, 34(5):1590 – 1603, 2016.

[18] S. J. Kazempour, A. . J. Conejo, and C. Ruiz. Generation investment equilibria with strategic producers – part i: Formulation. *Power Systems, IEEE Transactions on*, 28(3):2613 – 2622, 2013.

[19] F. Kong and X. Liu. GreenPlanning: Optimal Energy Source Selection and Capacity Planning for Green Datacenters. In *Proc. of ACM/IEEE International Conference on Cyber-Physical Systems*, 2016.

[20] S. Leyffer and T. Munson. Solving multi-leader-common-follower games. *Optimization Methods and Software*, (4):601–623, 2010.

[21] Z. Liu, I. Liu, S. Low, and A. Wierman. Pricing Data Center Demand Response. In *Proc. of ACM SIGMETRICS*, 2014.

[22] L. Lu, J. Tu, C. Chau, M. Chen, and X. Lin. Online Energy Generation Scheduling for Microgrids with Intermittent Energy Sources and Co-Generation. In *Proc. of ACM SIGMETRICS*, 2013.

[23] A. Qureshi. *Power-demand routing in massive geo-distributed systems*. PhD thesis, MIT, 2010.

[24] T. K. Ralphs. Parallel branch and cut. *Parallel Combinatorial Optimization*, pages 53 – 101, 2006.

[25] S. M. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, and J. Ousterhout. It's Time for Low Latency. In *Proc. of USENIX HotOS*, 2011.

[26] N. Tran, D. Tran, S. Ren, Z. Han, and C. Hong. How geo-distributed data centers do demand response: A game-theoretic approach. *Smart Grid, IEEE Transactions on*, 7(2):937 – 947, 2016.

[27] N. H. Tran, C. T. Do, S. Ren, Z. Han, and C.-S. Hong. Incentive mechanisms for economic and emergency demand responses of colocation datacenters. *IEEE Journal on Selected Areas in Communications*, 33(12):2892 – 2905, 2015.

[28] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad. Exploring Smart Grid and Data Center Interactions for Electric Power Load Balancings. In *Proc. of ACM GreenMetrics*, 2013.

[29] Y. Wang, X. Lin, and M. Pedram. A Stackelberg Game-Based Optimization Framework of the Smart Grid with Distributed PV Power Generations and Data Centers. In *Proc. of IEEE PES Innovative Smart Grid Technologies Conference*, 2013.

[30] S. Wogrin, J. Barquin, and E. Centeno. Capacity expansion equilibria in liberalized electricity markets: An epec approach. *Power Systems, IEEE Transactions on*, 28(2):1531 – 1539, 2013.

[31] H. Xu, C. Feng, and B. Li. Temperature Aware Workload Management in Geo-distributed Datacenters. In *Proc. of USENIX ICAC*, 2013.

[32] L. Zhang, Z. Li, C. Wu, and S. Ren. Online electricity cost saving algorithms for co-location data centers. *IEEE Journal on Selected Areas in Communications*, 33(12):2906 – 2919, 2015.

[33] L. Zhao, J. Brouwer, S. James, E. Peterson, D. Wang, and J. Liu. Fuel cell powered data centers: In-rack dc generation. *The Electrochemistry Society Transactions*, 77(1):131 – 139, 2016.

[34] B. Zhou, F. Zhang, L. Wang, C. Hou, A. F. Anta, A. V. Vasilakos, Y. Wang, J. Wu, and Z. Liu. Hdeer: A distributed routing scheme for energy-efficient networking. *IEEE Journal on Selected Areas in Communications*, 34(5):1713 – 1727, 2016.

[35] Z. Zhou, F. Liu, B. Li, B. Li, H. Jin, R. Zou, and Z. Liu. Fuel Cell Generation in Geo-Distributed Cloud Services: A Quantitative Study. In *Proc. of IEEE ICDCS*, 2014.

[36] Z. Zhou, F. Liu, and Z. Li. Pricing Bilateral Electricity Trade between Smart Grids and Hybrid Green Datacenters. In *Proc. of ACM SIGMETRICS*, 2015. Extended Abstract.

[37] Z. Zhou, F. Liu, Z. Li, and H. Jin. When Smart Grid Meets Geo-distributed Cloud: An Auction Approach to Datacenter Demand Response. In *Proc. of IEEE INFOCOM*, 2015.

[38] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, and H. Jin. Carbon-aware online control of geo-distributed cloud services. *Parallel and Distributed Systems, IEEE Transacttions on*, Preprint.

Zhi Zhou received the B.E. and M.E. degrees from the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), Wuhan, China. He is currently working toward the Ph.D. degree in the School of Computer Science and Technology, HUST. His primary research interests include green cloud computing, smart grid and wide-area big data analytics.

Fangming Liu (S'08-M'11-SM'16) received his B.Engr. degree in 2005 from Department of Computer Science and Technology, Tsinghua University, Beijing; and his Ph.D. degree in computer science and engineering from Hong Kong University of Science and Technology, Hong Kong, in 2011. He is a full professor in Huazhong University of Science and Technology, Wuhan, China. His research interests include cloud computing and datacenter, mobile cloud, green computing, SDN/NFV and virtualization. He is selected into National Program for Support of Top-Notch Young Professionals of National Program for Special Support of Eminent Professionals. He is a Youth Scientist of National 973 Basic Research Program Project of SDN-based Cloud Datacenter Networks. He was a StarTrack Visiting Faculty in Microsoft Research Asia in 2012-2013. He has been the Editor-in-Chief of EAI Endorsed Transactions on Collaborative Computing, a guest editor for IEEE Network Magazine, and served as TPC for ACM Multimedia 2014 and 2016, e-Energy 2016, IEEE INFOCOM 2013-2017, ICNP 2014 TPC and 2016 Poster/Demo Co-Chair, IWQoS 2016-2017 and ICDCS 2015-2016. He is a Senior Member of IEEE.

Zongpeng Li received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, in 1999, the M.S. degree in computer science, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2001 and 2005, respectively. Since August 2005, he has been with the Department of Computer Science, University of Calgary, Calgary, AB, Canada. From 2011 to 2012, he was a Visitor at the Institute of Network Coding, Chinese University of Hong Kong, Hong Kong. His research interest includes computer networks.