DGLB: Distributed Stochastic Geographical Load Balancing over Cloud Networks

Tianyi Chen, *Student Member, IEEE*, Antonio G. Marques, *Senior Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

Abstract—Contemporary cloud networks are being challenged by the rapid increase of user demands and growing concerns about global warming, due to their substantial energy consumption. This requires future data centers to be both energy efficient and sustainable, which calls for leveraging cutting-edge features and the flexibility provided by the modern smart grids. To fulfill those goals, this paper puts forward a systematic approach to designing energy-aware traffic-efficient geographical *load balancing schemes* for data-center networks that are not only optimal, but also computationally efficient and amenable to *distributed* implementation. Under this comprehensive approach, workload and power balancing schemes are designed jointly across the network, both delay-tolerant and *interactive workloads* are accommodated, novel smart-grid features such as energy storage units are incorporated to cope with renewables, and *incentive pricing* mechanisms are adopted in the design. To further account for the spatio-temporal variation of demands, energy prices and renewables, the task is formulated as a two-timescale stochastic optimization. Leveraging dual stochastic approximation and the fast iterative shrinkage-thresholding algorithm (FISTA), the proposed optimization is decomposed across time slots (first-stage) and data centers (second-stage). While the resultant online algorithm is strictly feasible and provably optimal under a Markovian assumption for the underlying random processes, extensive numerical tests further demonstrate that it also works well in real-data scenarios, where the underlying randomness is highly correlated across time.

Index Terms—Data center, renewables, energy storages, incentive payment, network resource allocation, stochastic programming

1 Introduction

The recent trend towards cloud computing has created a new class of computing systems, known as warehouse-scale computers, or, data centers (DCs) [1], which are rapidly proliferating all over the world. DCs are nowadays essential to provide Internet services such as web search, video distribution or data analytics. To enhance reliability and quality-of-service (QoS), cloud-service providers usually deploy DCs across different geographical regions. As an example, Apple is undertaking its biggest European DC project to date, with an investment of around $1.9 billion on two massive DCs, one in Ireland and one in Denmark [2]. Along with their growth in number and scale, the considerable amount of energy consumed by DCs not only challenges DC operators' budgets, but also raises global warming and climate-change concerns [3].

Optimal energy and workload management for setups with a *single* DC have been thoroughly investigated [4], [5], [6], [7], [8]. The issues of speed scaling and dynamic resizing in a server farm were considered in [4] and [5], while optimal power control with energy storage devices was discussed in [6]; however, [4], [5], [6] did not account for renewable energy sources (RES), which have been investigated separately in [9] and [10]. Schemes to minimize the cooling consumption when future state information is known were reported in [7]; see also [8] for stochastic formulation. However, [7] and [8] process user requests locally or presume that optimal routing has been performed, thus missing to leverage the spatio-temporal diversity of RES, data demand and energy prices.

Fewer works have dealt with the *geographical load balancing* task over a DC *network* [11], [12], [13], [14], [15]. To distribute delay-tolerant workloads (DWs), [11] developed an online algorithm with a desirable tradeoff among energy cost, workload fairness and latency. A Lyapunov-optimization-based approach was proposed in [12] for joint workload routing and thermal storage management for geo-distributed DCs, while [13] adopted a two-timescale scheme to solve the workload management at a fast timescale and the server activation at a slow timescale. However, the approaches in [11], [12], [13] were tailored to schedule DWs, and the generalization to interactive workloads (IWs) is not straightforward. To account for IWs, alternating direction method of multipliers (ADMM)-based schemes were employed in [14] and [15] to design distributed algorithms that solved the cost minimization over a single time slot. Yet, the approaches in [14] and [15] mainly deal with a deterministic workload allocation problem, rather than considering the spatio-temporal uncertainty inherent to data demands, energy prices and renewables, which challenges the design of schemes incorporating both DW and IW traffic.

- *T. Chen and G.B. Giannakis are with the Department of Electrical and Computer Engineering, and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455. E-mail: {chen3827, georgios}@umn.edu.*
- *A.G. Marques is with the Department of Signal Theory and Communications, Rey Juan Carlos University, Madrid 28943, Spain. E-mail: antonio.garcia.marques@urjc.es.*

Manuscript received 12 Mar. 2016; revised 15 Nov. 2016; accepted 3 Dec. 2016. Date of publication 6 Dec. 2016; date of current version 14 June 2017. Recommended for acceptance by B. Urgaonkar.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2016.2636210

Different from [4], [5], [6], [7], [8], [9], [10], the present paper deals with online joint workload and energy management for a cloud network consisting of multiple geo-distributed mapping nodes (MNs) and DCs. The intended resource allocation task is formulated as an infinite time horizon optimization task to minimize the time-average network-wide cost, subject to various operational constraints. Compared to [11], [12], [13], [14], [15], the proposed workload routing and scheduling policy considers *both DWs and IWs*, while the energy management scheme integrates renewables, storage units and two-way energy trading, to minimize the total energy cost from cooling and information technology (IT) operating systems. Furthermore, leveraging the flexibility provided by the new demand-response programs [16] and [17], a simple yet efficient incentive payment mechanism is introduced, which modulates the peaks of IW demand, while guaranteeing QoS; see also [15] and references therein. Major research challenges include: i) accounting for and exploiting the spatio-temporal variations of the state variables such as RES, IT demand and energy prices, even when their joint distribution is unknown; ii) joint consideration of costs and constraints that couple optimization variables across time and space; and iii) development of low-complexity distributed algorithms that can be implemented in real time.

The main contribution of this paper is the development of DGLB, a two-timescale algorithm for *online distributed geographical load balancing* over cloud networks. DGLB is obtained as the solution of a rigorously formulated nonlinear constrained optimization; it accounts for the inherent spatio-temporal uncertainty in the network (including renewables, energy prices and users demands); entails a low computational complexity; can be implemented distributedly; does not require statistical knowledge of the random variables involved; and has provable convergence and optimality. We summarize the contributions of this paper as follows.

C1) Targeting a unified framework for geographical load balancing in DCs, we take a holistic view and encompass a number of factors related to workload and energy management in a model *more comprehensive than* that adopted by state-of-the-art approaches. Compared to [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], online resource allocation incorporating both DWs and IWs is tackled, which challenges the development of fast and distributed solvers.

C2) We decouple the optimization variables across time by judicious relaxation, and reformulate the dynamic problem as a stationary stochastic program. Leveraging a two-stage Lagrange relaxation, we develop a novel two-timescale stochastic resource allocation scheme termed DGLB. Specifically, i) a stochastic dual gradient method is run at a slow timescale to deal with the long-term constraints; and ii) a diagonally weighted FISTA is run at a fast timescale to ensure fast real-time coordination and decentralized implementation. DBLG can operate without knowing the distribution of the involved random variables, and requires each agent in the network to collect information only from its one-hop neighbors.

C3) While most existing works in this setting (e.g., [6], [8], [12], [13]) adopt an i.i.d. assumption for the underlying random state variables, we analytically establish feasibility and optimality of our two-timescale online approach under a more general Markovian assumption, which readily includes the i.i.d. setting and can accommodate a wider range of real-world applications. We further demonstrate, by extensive simulations, that our DGLB also works well in more challenging real-world scenarios, where the underlying randomnesses are highly correlated over time.

Paper Outline. Modeling preliminaries are given in Section 2. The stochastic geographical management scheme is proposed in Section 3, while its distributed implementation is developed in Section 4. Performance analysis is presented in Section 5. Numerical tests are provided in Section 6, followed by conclusions in Section 7.

2 Modeling Preliminaries

Our system operates on discrete time slots indexed by t, with an infinite scheduling horizon $\mathcal{T} := \{0, 1, \ldots\}$. A network with $\mathcal{J} := \{1, 2, \ldots, J\}$ MNs, and $\mathcal{I} := \{1, 2, \ldots, I\}$ heterogeneous DCs is considered. MNs collect user requests over a geographical area (e.g., a city or a state) and forward the corresponding workloads to one or more DCs, which are distributed across a large area (e.g., a country). In addition to the IT system present to process the assigned workloads, each DC is equipped with a cooling system to remove the heat generated by the IT system, and a power supply system supporting the IT and cooling infrastructure. MNs make forwarding decisions based on the user requirements, the communication and networking costs, the load of different DCs, and their marginal energy price. The goal is to leverage the spatio-temporal variation of communication costs, energy prices, RES and cooling supplies, to obtain a more efficient network operation.

The ensuing sections describe the detailed operation of each MN and DC, including workload, network, power supply and power demand models, as well as the different system costs and incentive payment mechanisms that can be used to modulate the users' demand.

2.1 Traffic Workloads and Network Constraints

Suppose that each MN collects two types of workloads: delay-sensitive interactive and delay-tolerant workloads [13]. The IWs such as instant messaging and voice services are real-time requests that need to be served immediately. DWs are relatively time insensitive and deferrable within given slots. Typical examples include system updates and data backup. This provides ample optimization opportunities for workload allocation based on the dynamic variation of energy prices and RES availabilities. Table 1 summarizes all the notation introduced in this section.

For IWs, let $V_{j,t}$ denote the workload requested (arrival rate) to MN j at time t, and $v_{i,j,t}$ the amount of workload distributed from MN j to DC i at time t. Per slot t, MN j should dispatch all arrived IWs to a set of DCs physically connected to it. If $\mathcal{I}_j \subseteq \mathcal{I}$ denotes the set of DCs connected to MN j, the following constraints must be satisfied

$$\textstyle\sum_{i \in \mathcal{I}_j} v_{i,j,t} = V_{j,t}, \quad \forall j, t. \tag{1}$$

Although multiple IW types can be considered, since all must be served immediately, to simplify notation we aggregate them to $V_{j,t}$. In contrast, multiple types of DW are collected in the set $\mathcal{Q} := \{1, 2, \ldots, Q\}$. The reason for considering multiple classes of DWs is twofold: i) the utility generated by each of the services can be different, and ii) since this type of

TABLE 1
Notation List

Symbol	Definition
$i, I, \mathcal{I}$	Index, number, and set of DCs.
$j, J, \mathcal{J}$	Index, number, and set of MNs.
$q, Q, \mathcal{Q}$	Index, number, and set of DWs.
$V_{j,t}$	Amount of arrived IW at MN j per slot t.
$v_{i,j,t}$	Amount of IW routed from MN j to DC i per slot t.
$W_{j,q,t}$	Amount of arrived DW q at MN j per slot t.
$\tilde{w}_{i,j,q,t}$	Amount of DW q routed from MN j to DC i per slot t.
$w_{i,q,t}$	Amount of DW q being processed at DC i per slot t.
$Y^{\text{mn}}_{j,q,t}$	Queue length of DW q in MN j at the beginning of slot t.
$Y^{\text{dc}}_{i,q,t}$	Queue length of DW q in DC i at the beginning of slot t.
$d_{i,t}$	Total IT demand of DC i during slot t.
$B_{j,i}$	Limit of distribution rate per link from MN j to DC i.
$\overline{P}^{\text{s}}_i$	Peak power consumption of a server in DC i per slot.
M_i	Total number of servers in DC i.
D_i	Computing capacity of a single server in DC i.
$e_{i,t}$	Power coefficient reflecting environment factors.
$P^{\text{g}}_{i,t}, \overline{P}^{\text{g}}_i$	CG generation in DC i per slot t and its upper-limit.
$P^{\text{r}}_{i,t}, \overline{P}^{\text{r}}_i$	RG generation in DC i per slot t and its upper-limit.
$P^{\text{b}}_{i,t}$	Power (dis-)charged to the battery in DC i per slot t.
$\underline{P}^{\text{b}}_i, \overline{P}^{\text{b}}_i$	Lower- and upper-limits of $P^{\text{b}}_{i,t}$.
$Y^{\text{b}}_{i,t}$	SOC of the battery in DC i at the beginning of slot t.
$P^{\text{m}}_{i,t}$	Power that DC i buys/sells from/to the market at time t.
$\alpha^{\text{p}}_{i,t}, \alpha^{\text{s}}_{i,t}$	Price of buying/selling energy by DC i per slot t.
$p_{j,t}$	Price that MN j pays for demand curtailment at time t.
$\tilde{V}_{j,t}$	IW amount that users in MN j will reduce per slot t.
$\hat{V}_{j,t}$	Actual amount of IW arriving at MN j per slot t.
$P^{\text{s}}_i(\cdot)$	Power consumption of a single server in DC i per slot.
$P^{\text{it}}_i(\cdot)$	Power consumption of all servers in DC i per slot.
$P^{\text{ac}}_i(\cdot)$	Power consumption of cooling facilities in DC i per slot.
$U^{\text{v}}_j(\cdot)$	Revenue from IWs at MN j per slot.
$U^{\text{w}}_{i,q}(\cdot)$	Revenue from DW q at DC i per slot.
$G^{\text{d}}_{i,j}(\cdot)$	Cost of distributing loads from MN j to DC i per slot.
$G^{\text{e}}_i(\cdot)$	Energy transaction cost in DC i per slot.
$G^{\text{c}}_i(\cdot)$	Cost of CG in DC i per slot.
$G^{\text{b}}_i(\cdot)$	Cost of (dis-)charging the battery in DC i per slot.
$G^{\text{u}}_j(\cdot)$	Cost of user dissatisfaction at MN j per slot.

workloads is deferrable, the developed algorithms can give different priority to each of the services. In this case, let $W_{j,q,t}$ and $\tilde{w}_{i,j,q,t}$ denote the amount of DW q arriving at MN j at slot t and the amount of DW q routed from MN j to DC i at slot t, respectively. Since DWs are deferrable, the fraction of unrouted workload is buffered in queues (one per class of DW) obeying the following dynamic recursion

$$Y^{\text{mn}}_{j,q,t+1} = \left[Y^{\text{mn}}_{j,q,t} + W_{j,q,t} - \textstyle\sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\right]^{\infty}_{0}, \ \forall j,q,t, \qquad (2)$$

where $Y^{\text{mn}}_{j,q,t}$ is the queue length of DW q in MN j at the beginning of slot t, and $[\cdot]^b_a := \max\{a, \min\{b, \cdot\}\}$.

At the DC side, IWs must be processed once received, while DWs are deferrable. With $w_{i,q,t}$ denoting the amount of DW q processed by DC i during slot t, the unserved portion of the workloads are buffered at the DC using separate queues. This leads to the following dynamic recursion

$$Y^{\text{dc}}_{i,q,t+1} = \left[Y^{\text{dc}}_{i,q,t} - w_{i,q,t} + \textstyle\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t}\right]^{\infty}_{0}, \ \forall i,q,t, \qquad (3)$$

where $Y^{\text{dc}}_{i,q,t}$ is the queue length of DW q in DC i at the beginning of slot t. Queue dynamics slightly different from the one in (2) and (3) can also be considered [12], [13], but such differences are not relevant for the subsequent analysis.

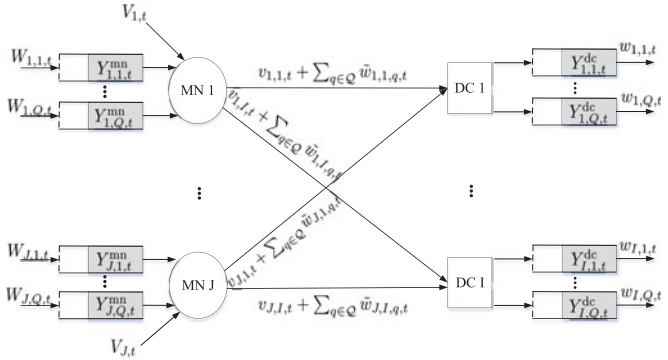

Fig. 1. A geographical load balancing system diagram.

The total IT demand of DC i in slot t, is thus the superposition of IWs and DWs, which is given by

$$d_{i,t} = \textstyle\sum_{j\in\mathcal{J}} v_{i,j,t} + \sum_{q\in\mathcal{Q}} w_{i,q,t}, \quad \forall i,t. \qquad (4)$$

Lastly, to account for the bandwidth of the MN-to-DC links, the total workload distribution rate per link from MN j to DC i is upper bounded by the time-invariant constant

$$v_{i,j,t} + \textstyle\sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t} \le B_{j,i}, \quad \forall j,i,t, \qquad (5)$$

with $B_{j,i} = 0$ if MN j and DC i are not connected. To simplify derivations, each MN-to-DC link is modeled here as a single-hop channel with a dedicated bandwidth constraint. However, the proposed formulation can be modified to accommodate multi-hop communications among MNs and DCs. Although this will require incorporation of routing variables and flow conservation constraints, the basic structure of the problem and the approach to solve it will remain the same; see, e.g., [18] for details.

A workload allocation diagram summarizing the variables introduced in Section 2.1 is presented in Fig. 1.

2.2 Power Demand and Supply Models

The main cost when operating a DC is due to its power consumption. In this section, we describe the relation between the load served by a DC, and the corresponding power consumption, as well as the different sources of energy available at each DC.

We start by modeling P^{s}_i, the power consumed by a single server in DC i. Let $\overline{P}^{\text{s}}_i$ denote the peak power consumption of a server in DC i, and let $c \in [0,1]$ denote the speed of the server, oftentimes referred to as CPU usage (processed work divided by the server capacity). Power P^{s}_i can be then approximated as $P^{\text{s}}_i(c) = \overline{P}^{\text{s}}_i(\varrho c^{\sigma} + 1 - \varrho)$, where the fraction of peak consumption $1-\varrho$ represents the power consumed in idle state (i.e., $c = 0$), which is around 0.4, and constant $\sigma \ge 1$ is typically set to 2 in state-of-the-art servers [4]. Assume also that the M_i servers in DC i are all identical. Then, given the total IT demand $d_{i,t}$ in DC i at time t, it follows from the convexity of $P^{\text{s}}_i(c)$ that the most energy-efficient allocation is to divide $d_{i,t}$ uniformly across servers. In this way, with D_i denoting the per-server capacity of DC i, all servers are running at the same speed $(d_{i,t}/M_i)/\ D_i \in [0,1]$, and the total power consumption at DC i can be expressed as

$$P_i^{\mathrm{it}}(d_{i,t}) = M_i P_i^{\mathrm{s}}\left(\frac{d_{i,t}}{M_i D_i}\right) = \frac{\varrho d_{i,t}^2}{M_i D_i^2}\overline{P}_i^{\mathrm{s}} + (1-\varrho) M_i \overline{P}_i^{\mathrm{s}}. \quad (6)$$

Clearly, function $P_i^{\mathrm{it}}(\cdot)$ is increasing and convex with respect to (w.r.t.) $d_{i,t}$. Here, the number of active servers M_i is assumed to be the same across the scheduling horizon. The reason for this is that the so-termed "switching cost" incurred from toggling a server in and out of a power-saving mode (including the delay, energy, and wear-and-tear costs) is substantial, so that frequently changing M_i is not beneficial. Additional details as well as specific research on dynamic sizing of DCs can be found in [5].

Along with the increasing density of IT equipment in DCs, a considerable amount of electricity is consumed by the cooling system [3]. We assume for simplicity that the cooling consumption is proportional to the total IT power consumption as $P_i^{\mathrm{ac}}(d_{i,t}) = e_{i,t} P_i^{\mathrm{it}}(d_{i,t})$, where $e_{i,t}$ is time-varying and depends on a variety of environment factors (e.g., humidity, temperature). A typical value is around 0.3 with advanced cooling facilities [7]. In any case, we will assume henceforth that *at time t* the value of $e_{i,t}$ is deterministically known. Note finally that although a simple cooling consumption model is adopted, our framework can easily include more advanced convex cooling consumption models; see, e.g., [7], [8].

The next step is to describe the power supply model. In particular, we assume that each DC is supplied by a renewable-integrated (micro-)grid consisting of a conventional generator (CG) (e.g., a fuel generator), an on-site renewable generator (RG) (e.g., wind or solar), and an energy storage unit (e.g., a battery). Specifically,

- $P_{i,t}^{\mathrm{g}}$ stands for the energy generated at time t by the CG in DC i, which is upper bounded by $\overline{P}_i^{\mathrm{g}}$, so that

$$0 \le P_{i,t}^{\mathrm{g}} \le \overline{P}_i^{\mathrm{g}}, \quad \forall t. \quad (7)$$

- $P_{i,t}^{\mathrm{r}}$ is the renewable energy generated at the beginning of slot t by the RG in DC i, which is also bounded in $0 \le P_{i,t}^{\mathrm{r}} \le \overline{P}_i^{\mathrm{r}}, \forall t$.
- $P_{i,t}^{\mathrm{b}}$ is the power delivered to or drawn from the battery (storage unit) in DC i at slot t, which amounts to either charging ($P_{i,t}^{\mathrm{b}} > 0$) or discharging ($P_{i,t}^{\mathrm{b}} < 0$) the battery. Let $Y_{i,0}^{\mathrm{b}}$ and $Y_{i,t}^{\mathrm{b}}$ denote the initial amount of stored energy and the state of charge of the storage unit in DC i at the beginning of time slot t. Each unit has a finite capacity $\overline{Y}_i^{\mathrm{b}}$ as well as a minimum level $\underline{Y}_i^{\mathrm{b}}$. The dynamics of the storage unit are described as

$$\underline{Y}_i^{\mathrm{b}} \le Y_{i,t}^{\mathrm{b}} \le \overline{Y}_i^{\mathrm{b}}, \quad \forall i,t \quad (8)$$

$$Y_{i,t+1}^{\mathrm{b}} = Y_{i,t}^{\mathrm{b}} + P_{i,t}^{\mathrm{b}}, \quad \forall i,t \quad (9)$$

$$\underline{P}_i^{\mathrm{b}} \le P_{i,t}^{\mathrm{b}} \le \overline{P}_i^{\mathrm{b}}, \quad \forall i,t, \quad (10)$$

  where the bounds on the (dis)charging amount $\underline{P}_i^{\mathrm{b}} < 0$ and $\overline{P}_i^{\mathrm{b}} > 0$ in (10) are dictated by physical limits.

In addition to the energy resources within the microgrid, the DCs can resort to the external wholesale electricity market in an on-demand manner. To be specific, $P_{i,t}^{\mathrm{m}}$ denotes the energy that DC i buys from the *market* at time t. Since a two-way energy trading facility is considered, if negative, $P_{i,t}^{\mathrm{m}}$ denotes the energy sold by the DC.

With these notational conventions, at each time t, the power demand and supply at each of the DCs has to be balanced. Mathematically, this amounts to requiring

$$P_{i,t}^{\mathrm{m}} + P_{i,t}^{\mathrm{g}} + P_{i,t}^{\mathrm{r}} = P_i^{\mathrm{it}}(d_{i,t}) + P_i^{\mathrm{ac}}(d_{i,t}) + P_{i,t}^{\mathrm{b}}. \quad (11)$$

Under constraints (7), (8), (9), (10), and (11), $P_{i,t}^{\mathrm{r}}$ is the state variable, while $\{P_{i,t}^{\mathrm{g}}, P_{i,t}^{\mathrm{b}}, P_{i,t}^{\mathrm{m}}\}$ are optimization variables.

2.3 Revenues and Operation Costs

Starting with the service and distribution of the workloads, we consider the *revenue* for IWs at MN j to be given by an increasing and *concave* utility function $U_j^{\mathrm{v}}(\cdot)$, and the revenue for DW q at DC i to be given by the increasing and concave utility $U_{i,q}^{\mathrm{w}}(\cdot)$. On the other hand, distribution of workloads across the network generates bandwidth *costs*. To this end, we will use the convex function $G_{i,j}^{\mathrm{d}}(\cdot)$ to denote the cost of distributing workloads from MN j to DC i, which, among other factors, will depend on the distance between them.

Regarding power supply sources, each DC can buy energy from external energy markets in period t at price $\alpha_{i,t}^{\mathrm{p}}$ (if $P_{i,t}^{\mathrm{m}} > 0$), or, sell energy to the markets at price $\alpha_{i,t}^{\mathrm{s}}$ if ($P_{i,t}^{\mathrm{m}} < 0$). Clearly, the shortage energy that needs to be purchased by the DC is $[P_{i,t}^{\mathrm{m}}]^+$; while the surplus energy that can be sold is $[P_{i,t}^{\mathrm{m}}]^-$. Notwithstanding, we shall always consider that $\alpha_{i,t}^{\mathrm{p}} \ge \alpha_{i,t}^{\mathrm{s}}$. This prevents less relevant buy-and-sell activities of the DC for profit and guarantees that, for every t, either the shortage or the surplus energy is zero, so that at most one of them can be positive. Those prices can be used to define the *energy transaction cost* between the DC microgrid and the external market per time t

$$G_i^{\mathrm{e}}(P_{i,t}^{\mathrm{m}}) := \alpha_{i,t}^{\mathrm{p}}[P_{i,t}^{\mathrm{m}}]^+ - \alpha_{i,t}^{\mathrm{s}}[P_{i,t}^{\mathrm{m}}]^-. \quad (12)$$

Moreover, we will use the convex function $G_i^{\mathrm{c}}(\cdot)$ to denote the *cost* of CG during time t, which typically is smooth quadratic [19]. Finally, to model the potential battery degeneration during the charging/discharging cycle, a strongly convex (dis)charging cost $G_i^{\mathrm{b}}(\cdot)$ can be employed to prevent fast and frequent (dis)charging of batteries [20].

2.4 Incentive Payment Models

While most existing works (e.g., [5], [7], [8], [14]) assume that IWs are fixed and inelastic, a number of interactive services tolerate their partial execution [15] (a.k.a. workload curtailment). This motivates MNs to offer incentive prices for end-users to curtail their instantaneous demand or accept partial execution, so that the peak demand is reduced under the guaranteed QoS [21]. These incentive prices are usually offered when the local marginal price is high, or when the grid operator sends emergency demand response signals such as a power outage [16].

Mathematically, let $p_{j,t}$ denote the incentive price that at time t MN j pays to users willing to reduce their IW [17]. This way, if users reduce their demand by an amount $\check{V}_{j,t}$,

the MN will pay $p_{j,t}\check{V}_{j,t}$. It is further assumed that users react to the given price in a simple non-strategic manner. Specifically, when *reducing their workload demand by* an amount $\check{V}_{j,t}$, users incur a strictly convex unsatisfactory cost $G_j^{\mathrm{u}}(\check{V}_{j,t}) = \kappa_j(\check{V}_{j,t})^2$, where the coefficient $\kappa_j > 0$ can be learned from historical data. Rational users set then their demand by solving the following optimization problem

$$\max_{0\leq \check{V}_{j,t}\leq \eta V_{j,t}} p_{j,t}\check{V}_{j,t} - G_j^{\mathrm{u}}(\check{V}_{j,t}), \tag{13}$$

where $V_{j,t}$ is the total interactive workload demand for MN j without incentive payment [cf. (1)], and η is the threshold of maximum workload reduction.

Note that (13) admits a closed-form solution, namely

$$\check{V}_{j,t} = \check{V}_j(p_{j,t}) = (\nabla G_j^{\mathrm{u}})^{-1}(p_{j,t}) = \left[p_{j,t}/2\kappa_j\right]_0^{\eta V_{j,t}}, \tag{14}$$

where ∇G_j^{u} denotes the gradient of G_j^{u} w.r.t. $\check{V}_{j,t}$, $(\nabla G_j^{\mathrm{u}})^{-1}$ is the inverse function of ∇G_j^{u}, and $[\,\cdot\,]_0^{\eta V_{j,t}}$ stands for the projection onto the interval $[0, \eta V_{j,t}]$. Hence, for a given incentive price $p_{j,t}$, the actual workload demand of MN j becomes $\tilde{V}_{j,t} = V_{j,t} - \check{V}_j(p_{j,t})$, which can be written as a convex (linear) function of $p_{j,t}$ as $\tilde{V}_{j,t} = \left[V_{j,t} - p_{j,t}/(2\kappa_j)\right]_{(1-\eta)V_{j,t}}^{\infty}$. As a result, the IW revenue $U_j^{\mathrm{v}}(\tilde{V}_{j,t})$ can written as $U_j^{\mathrm{v}}(p_{j,t}; V_{j,t})$; i.e., a function of $p_{j,t}$ parameterized by $V_{j,t}$.

3 Stochastic Load Balancing

Section 2 identified the variables, costs and constraints that must be accounted for in our network optimization problem, which is rigorously formulated here. In particular, we aim to pursue online energy and workload management for the considered MN-DC network. At each time t, the system operator in each DC and MN performs real-time scheduling to optimize routing $\{v_{i,j,t}, \tilde{w}_{i,j,q,t}\}$, workloads $\{w_{i,q,t}\}$, DC data demand $\{d_{i,t}\}$, incentive prices $\{p_{j,t}\}$, CG generation $\{P_{i,t}^{\mathrm{g}}\}$, battery charging energy $\{P_{i,t}^{\mathrm{b}}\}$, and external power supply $\{P_{i,t}^{\mathrm{m}}\}$. The goal is to minimize the limiting average network cost, subject to IT operational constraints, as well as CG and storage constraints. It is instructive to collect all sources of randomness into the state vector $\boldsymbol{s}_t := \{\alpha_{i,t}^{\mathrm{p}}, \alpha_{i,t}^{\mathrm{s}}, W_{j,q,t}, V_{j,t}, P_{i,t}^{\mathrm{r}},\ \forall i,j,q\}$, and also all the optimization variables into $\mathbf{x}_t := \{v_{i,j,t}, w_{i,q,t}, \tilde{w}_{i,j,q,t}, d_{i,t}, p_{j,t},\ P_{i,t}^{\mathrm{m}}, P_{i,t}^{\mathrm{g}}, P_{i,t}^{\mathrm{b}},\ \forall i,j,q\}$. Strictly speaking, not all variables in $\mathbf{x}_t$ are free. Indeed, using some of the equality constraints in Section 2, the value of variables such as $d_{i,t}$ can be found using $v_{i,j,t}$ and $w_{i,q,t}$ via (4). The reason for writing them as optimization variables and forcing the equality through a constraint, which will also turn out out to facilitate distributed solvers, will be apparent later. The resultant aggregated *network cost* for the considered MN-DC network *at time t* is

$$\begin{aligned}\Psi_t = \Psi(\mathbf{x}_t; \boldsymbol{s}_t) := &\sum_{i\in\mathcal{I}}\Big(G_i^{\mathrm{e}}(P_{i,t}^{\mathrm{m}}) + G_i^{\mathrm{c}}(P_{i,t}^{\mathrm{g}}) + G_i^{\mathrm{b}}(P_{i,t}^{\mathrm{b}})\Big)\\ &- \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}} U_{i,q}^{\mathrm{w}}(w_{i,q,t}) + \sum_{j\in\mathcal{J}}\Big(p_{j,t}\check{V}_j(p_{j,t}) - U_j^{\mathrm{v}}(p_{j,t}; V_{j,t})\Big)\\ &+ \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} G_{i,j}^{\mathrm{d}}\Big(v_{i,j,t} + \sum_{q\in\mathcal{Q}}\tilde{w}_{i,j,q,t}\Big).\end{aligned} \tag{15}$$

Defining also $\mathbf{Y}_t := \{Y_{i,t}^{\mathrm{b}}, Y_{j,q,t}^{\mathrm{mn}}, Y_{i,q,t}^{\mathrm{dc}},\ \forall i,j,q\}$, the optimal scheduling is obtained as the solution to the following *long-term* network-optimization problem:

$$\Psi^* := \min_{\{\mathbf{x}_t, \mathbf{Y}_t,\ \forall t\}} \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\Psi(\mathbf{x}_t; \boldsymbol{s}_t)] \tag{16a}$$

$$\text{s.t.}\quad Y_{i,t+1}^{\mathrm{b}} = Y_{i,t}^{\mathrm{b}} + P_{i,t}^{\mathrm{b}},\ \forall i,t \tag{16b}$$

$$\underline{Y}_i^{\mathrm{b}} \leq Y_{i,t}^{\mathrm{b}} \leq \overline{Y}_i^{\mathrm{b}},\ \forall i,t \tag{16c}$$

$$\underline{P}_i^{\mathrm{b}} \leq P_{i,t}^{\mathrm{b}} \leq \overline{P}_i^{\mathrm{b}},\ \forall i,t \tag{16d}$$

$$0 \leq P_{i,t}^{\mathrm{g}} \leq \overline{P}_i^{\mathrm{g}},\ \forall i,t \tag{16e}$$

$$0 \leq d_{i,t} \leq M_i D_i,\ \forall i,t \tag{16f}$$

$$\sum_{i\in\mathcal{I}_j} v_{i,j,t} = V_{j,t} - p_{j,t}/(2\kappa_j),\ \forall j,t \tag{16g}$$

$$0 \leq p_{j,t} \leq 2\eta\kappa_j V_{j,t},\ \forall j,t \tag{16h}$$

$$v_{i,j,t} + \sum_{q\in\mathcal{Q}}\tilde{w}_{i,j,q,t} \leq B_{j,i},\ \forall j,i \tag{16i}$$

$$d_{i,t} = \sum_{j\in\mathcal{J}} v_{i,j,t} + \sum_{q\in\mathcal{Q}} w_{i,q,t},\ \forall i,t \tag{16j}$$

$$P_{i,t}^{\mathrm{m}} + P_{i,t}^{\mathrm{g}} + P_{i,t}^{\mathrm{r}} = P_i^{\mathrm{it}}(d_{i,t}) + P_i^{\mathrm{ac}}(d_{i,t}) + P_{i,t}^{\mathrm{b}},\ \forall i,t \tag{16k}$$

$$Y_{j,q,t+1}^{\mathrm{mn}} = \Big[Y_{j,q,t}^{\mathrm{mn}} + W_{j,q,t} - \sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q,t}\Big]_0^{\infty},\ \forall j,q,t \tag{16l}$$

$$Y_{i,q,t+1}^{\mathrm{dc}} = \Big[Y_{i,q,t}^{\mathrm{dc}} - w_{i,q,t} + \sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}\Big]_0^{\infty},\ \forall i,q,t \tag{16m}$$

$$Y_{j,q,t}^{\mathrm{mn}} < \infty,\ \forall j,q,t;\ Y_{i,q,t}^{\mathrm{dc}} < \infty,\ \forall i,q,t, \tag{16n}$$

where the objective considers all time instants jointly (i.e., the entire scheduling horizon), and the expectation is taken over all sources of randomness (i.e., all variables in $\boldsymbol{s}_t$). Two additional remarks on (16) are in order. First, feasibility of (16) is assumed throughout the paper. However, some of the constraints could render the problem infeasible, and this could be detected by tracking the corresponding multipliers. Second, although strictly speaking the problem in (16) is convex, the battery dynamics in (16b) as well as the delay-tolerant workload queues in (16l) and (16m) couple the optimization variables over the infinite time horizon. This requires running a joint optimization scheme with a prohibitively high dimensionality. Even worse, for the practical case where the knowledge of $\boldsymbol{s}_t$ is causal, finding the optimal solution while accounting for the coupling across time calls for dynamic programming tools, which are generally intractable. Our approach to circumventing this obstacle is to relax (16b), (16l) and (16m), by replacing them with average constraints, and employ dual decomposition techniques to separate the solution across time. This is elaborated in the next section.

3.1 Problem Relaxation

Combining (16l), (16m) and (16n), it follows that in the long term the average workload arrival and departure rates must satisfy the following necessary conditions

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[W_{j,q,t}\big]\leq\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q,t}\right],\ \forall j,q \quad (17a)$$

and

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}\right]\leq\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[w_{i,q,t}\big],\ \forall i,q. \quad (17b)$$

In words, in the long term all buffered DWs should be served. Upon observing that the batteries in (16b) and (16c) exhibit dynamics very similar to those of the workload queues, we use the same relaxation and require

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\Big[P_{i,t}^{\mathrm{b}}\Big]=0,\quad\forall i. \quad (17c)$$

As before, (17c) guarantees that in the long term the energy stored into the battery and the energy taken from it are equal. Using (17a), (17b), and (17c), a relaxed version of (16) is

$$\begin{aligned}\tilde{\Psi}^{*}:=&\min_{\{\mathbf{x}_t\}}\ \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\Psi(\mathbf{x}_t;\boldsymbol{s}_t)]\\ &\text{s.t.}\quad (16\text{d})-(16\text{j}),\ (17\text{a})-(17\text{c}).\end{aligned} \quad (18)$$

Compared to (16), variables $\mathbf{\Upsilon}_t:=\{Y_{i,t}^{\mathrm{b}},Y_{j,q,t}^{\mathrm{mn}},Y_{i,q,t}^{\mathrm{dc}},\forall i,j,q\}$ are not present in (18), and the time-coupling constraints (16b) and (16l) and (16m) are replaced with (17a), (17b), and (17c).

The problem in (18) has a number of interesting properties, including: a) since (18) is a relaxed version of (16), it follows that $\tilde{\Psi}^{*}\leq\Psi^{*}$; b) if $\{\boldsymbol{s}_t\}$ is stationary, the solution is stationary too and easy to characterize—this will be further discussed in the next paragraph; and c) as will argued in Section 5, there exist low-complexity solvers that approximate the solution of (18) while being feasible for (16).

Regarding property b), using arguments similar to those in, e.g., [6], [22], it can be shown that if the random process $\{\boldsymbol{s}_t\}$ is stationary, there exists a *time-invariant* control policy $\mathbf{x}:\boldsymbol{s}_t\to\mathbf{x}_t$ inducing a solution $\mathbf{x}_t=\mathbf{x}_t(\boldsymbol{s}_t)=\mathbf{x}(\boldsymbol{s}_t)$, which can be shown to: satisfy the constraints (16d), (16e), (16f), (16g), (16h), (16i), and (16j); achieve optimal performance $\mathbb{E}[\Psi(\mathbf{x}(\boldsymbol{s}_t);\boldsymbol{s}_t)]=\tilde{\Psi}^{*}$; and satisfy the long-term constraints $\mathbb{E}[W_{j,q}(\boldsymbol{s}_t)-\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q}(\boldsymbol{s}_t)]\leq 0,\ \forall j,q$, $\mathbb{E}[w_{i,q}(\boldsymbol{s}_t)-\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q}(\boldsymbol{s}_t)]\leq 0,\ \forall i,q$ and $\mathbb{E}[P_i^{\mathrm{b}}(\boldsymbol{s}_t)]=0,\ \forall i$ [cf. (17a), (17b), and (17c)]. A critical implication of this is that all the expectations inside the limiting time averages in (18) yield the same result. Therefore, the time averages can be removed and the problem can be tackled using "standard" convex stochastic programming tools. To handle the coupling across optimization variables introduced by the expectations in (17a), (17b), and (17c), we will dualize the long-term constraints (17a), (17b), and (17c), and use a decomposition approach in the dual domain. As explained in detail in the next section, after the dualization, the optimal solution for each t can be computed separately across time.

3.2 Dual Decomposition

Let $\{\lambda_{j,q}^{\mathrm{mn}}\}$, $\{\lambda_{i,q}^{\mathrm{dc}}\}$ and $\{\lambda_i^{\mathrm{b}}\}$ denote the Lagrange multipliers associated with constraints (17a), (17b), and (17c), respectively. With $\mathbf{x}_{\mathcal{T}}:=\{\mathbf{x}_t\}_{t\in\mathcal{T}}=\{\mathbf{x}(\boldsymbol{s}_t)\}_{t\in\mathcal{T}}$, and $\boldsymbol{\lambda}$ collecting all the multipliers, the partial Lagrangian function of (18) is

$$\begin{aligned}\mathcal{L}(\mathbf{x}_{\mathcal{T}},\boldsymbol{\lambda}):=&\,\mathbb{E}[\Psi(\mathbf{x}(\boldsymbol{s}_t);\boldsymbol{s}_t)]+\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}\mathbb{E}\left[\lambda_{i,q}^{\mathrm{dc}}\Big(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q}(\boldsymbol{s}_t)-w_{i,q}(\boldsymbol{s}_t)\Big)\right]\\ &+\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\mathbb{E}\left[\lambda_{j,q}^{\mathrm{mn}}\Big(W_{j,q}(\boldsymbol{s}_t)-\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q}(\boldsymbol{s}_t)\Big)\right]\\ &+\sum_{i\in\mathcal{I}}\mathbb{E}\big[\lambda_i^{\mathrm{b}}(P_i^{\mathrm{b}}(\boldsymbol{s}_t))\big],\end{aligned} \quad (19)$$

where the expectation is taken over the steady-state distribution of $\{\boldsymbol{s}_t\}$. Since this distribution is time-invariant, note that the index t in $\boldsymbol{s}_t$ could be dropped.

With $\mathcal{X}_t:=\mathcal{X}(\mathbf{s}_t)$ denoting feasible set defined by the instantaneous constraints (16d), (16e), (16f), (16g), (16h), (16i), and (16j), which are the ones not dualized in (19), the Lagrange dual function is

$$\mathcal{D}(\boldsymbol{\lambda}):=\min_{\{\mathbf{x}(\mathbf{s}_t)\in\mathcal{X}(\mathbf{s}_t)\}_{t\in\mathcal{T}}}\ \mathcal{L}(\mathbf{x}_{\mathcal{T}},\boldsymbol{\lambda}) \quad (20)$$

and the dual problem of (18) is

$$\max_{\boldsymbol{\lambda}}\ \mathcal{D}(\boldsymbol{\lambda}). \quad (21)$$

For the dual problem (21), a standard iterative *dual subgradient algorithm* (DSA) can be employed to obtain the optimal $\boldsymbol{\lambda}^{*}$. Namely, with k being an iteration index, the multipliers at iteration $k+1$, denoted by $\boldsymbol{\lambda}(k+1)$, are found as

$$\lambda_{i,q}^{\mathrm{dc}}(k+1)\ =\big[\lambda_{i,q}^{\mathrm{dc}}(k)+\mu g_{\lambda_{i,q}^{\mathrm{dc}}}(k)\big]_0^{\infty},\ \forall i,q \quad (22a)$$

$$\lambda_{j,q}^{\mathrm{mn}}(k+1)=\big[\lambda_{j,q}^{\mathrm{mn}}(k)+\mu g_{\lambda_{j,q}^{\mathrm{mn}}}(k)\big]_0^{\infty},\ \forall j,q \quad (22b)$$

$$\lambda_i^{\mathrm{b}}(k+1)\ =\ \lambda_i^{\mathrm{b}}(k)+\mu g_{\lambda_i^{\mathrm{b}}}(k),\ \forall i, \quad (22c)$$

where $\mu>0$ is a constant stepsize that, if convenient, can be rendered different for each multiplier, and $g_{\boldsymbol{\lambda}}(k):=\{g_{\lambda_i^{\mathrm{b}}}(k),g_{\lambda_{i,q}^{\mathrm{dc}}}(k),g_{\lambda_{j,q}^{\mathrm{mn}}}(k),\ \forall i,j,q\}$ denote the subgradients of $\mathcal{D}(\boldsymbol{\lambda})$ in (20) w.r.t. the corresponding dual variables. These can be expressed as $g_{\lambda_{i,q}^{\mathrm{dc}}}(k)=\mathbb{E}\big[\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q}(\boldsymbol{s}_t;k)-w_{i,q}(\boldsymbol{s}_t;k)\big]$, $g_{\lambda_{j,q}^{\mathrm{mn}}}(k)=\mathbb{E}\big[W_{j,q}(\boldsymbol{s}_t;k)-\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q}(\boldsymbol{s}_t;k)\big]$, and, $g_{\lambda_i^{\mathrm{b}}}(k)=\mathbb{E}\big[P_i^{\mathrm{b}}(\boldsymbol{s}_t;k)\big]$, with $\mathbf{x}_{\mathcal{T}}(k)=\{\mathbf{x}_t(k)\}_{t\in\mathcal{T}}=\{\mathbf{x}(\boldsymbol{s}_t;k)\}_{t\in\mathcal{T}}$ standing for the primal minimizers of the Lagrangian for the kth iteration of the subgradient method, i.e., $\mathbf{x}_{\mathcal{T}}(k):=\arg\min_{\mathbf{x}_{\mathcal{T}}}\mathcal{L}(\mathbf{x}_{\mathcal{T}},\boldsymbol{\lambda}(k))$ subject to (16d), (16e), (16f), (16g), (16h), (16i), and (16j).

Due to the linearity of the $\mathbb{E}[\cdot]$ operator, the minimization w.r.t. the primal variables in (20) can be performed *separately across time*. Hence, the primal minimizers $\mathbf{x}_{\mathcal{T}}(k)=\{\mathbf{x}(\boldsymbol{s}_t;k)\}_{t\in\mathcal{T}}$ can be found by solving the following (infinitely many) instantaneous sub-problems (one per $\mathbf{s}_t$)

$$\begin{aligned}\mathbf{x}(\boldsymbol{s}_t;k)\in\arg\min_{\mathbf{x}_t}\ &\Psi(\mathbf{x}_t;\boldsymbol{s}_t)+\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}\lambda_{i,q}^{\mathrm{dc}}(k)\Big(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}\\ &-w_{i,q,t}\Big)+\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\lambda_{j,q}^{\mathrm{mn}}(k)\Big(W_{j,q,t}-\sum_{i\in\mathcal{I}}\tilde{w}_{i,j,q,t}\Big)+\sum_{i\in\mathcal{I}}\lambda_i^{\mathrm{b}}(k)P_{i,t}^{\mathrm{b}}\\ &\text{s.t.}\quad (16\text{d})-(16\text{j}),\end{aligned} \quad (23)$$

where the operator $\in$ accounts for cases that the Lagrangian has more than one minimizer. The problem in (23) is convex and has a small-to-moderate dimensionality, so that in most practical cases, it is not difficult to solve. In fact, for a number of relevant cost and utility functions (including quadratic and logarithmic), closed-form solutions for many of the primal variables can be found.

3.3 Stochastic DSA

The standard DSA in (22) involves taking the expectation over the stationary distribution of $\boldsymbol{s}_t$ to obtain the subgradient $g_{\boldsymbol{\lambda}}(k)$. This can be challenging not only for numerical reasons, but also because such distributions can be difficult to characterize or estimate when unknown. To circumvent this challenge, we will resort to stochastic approximation [23]. The benefits are multiple, including: a) considerably reduced computational complexity; b) the distribution of $\boldsymbol{s}_t$ need not be known; and, c) the resultant algorithms are robust to noise and non-stationary environments. Specifically, the iterations in (22) are replaced with

$$\lambda^{\mathrm{b}}_{i,t+1} = \lambda^{\mathrm{b}}_{i,t} + \mu P^{\mathrm{b}}_{i,t},\ \forall i \tag{24a}$$

$$\lambda^{\mathrm{dc}}_{i,q,t+1} = \left[\lambda^{\mathrm{dc}}_{i,q,t} + \mu\Big(\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} - w_{i,q,t}\Big)\right]^{\infty}_{0},\ \forall i,q \tag{24b}$$

$$\lambda^{\mathrm{mn}}_{j,q,t+1} = \left[\lambda^{\mathrm{mn}}_{j,q,t} + \mu\Big(W_{j,q,t} - \sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\Big)\right]^{\infty}_{0},\ \forall j,q, \tag{24c}$$

where $\{P^{\mathrm{b}}_{i,t}, \tilde{w}_{i,j,q,t}, w_{i,q,t}\}$ are found by solving

$$\begin{aligned}\min_{\mathbf{x}_t}\ \Phi(\mathbf{x}_t;\boldsymbol{s}_t) := \Psi(\mathbf{x}_t;\boldsymbol{s}_t) + \sum_{i\in\mathcal{I}}\Bigg[\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{dc}}_{i,q,t}\Big(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t} - w_{i,q,t}\Big) \\ + \lambda^{\mathrm{b}}_{i,t}P^{\mathrm{b}}_{i,t}\Bigg] + \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{mn}}_{j,q,t}\Big(W_{j,q,t} - \sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q,t}\Big) \\ \text{s.t.}\quad (16\mathrm{d})-(16\mathrm{j}).\end{aligned} \tag{25}$$

As will be shown in Section 5, the stochastic iterations in (24) and (25) come with two additional benefits critical for the problem at hand. First, there are performance and feasibility guarantees establishing that the solution provided by (25) is a tight approximation to the solution of (18). Second, if properly initialized, the solution to (25) can be shown to be feasible for the original problem in (16). Last but not least, links between the stochastic estimates in (24) and the battery and queue lengths can be established; see [24] and [25] for a rigorous discussion.

Remark 1. In practice, it can be useful to re-scale the subgradient in (24) so that each dual variable is updated with a different stepsize. First, if the order of magnitude of the battery (dis)charging and the workload arrival rate are very different, stepsize adjustment facilitates numerical convergence. Second, within one class of constraints—e.g., flow conservation at the MN side—*using different stepsizes* offers as a mechanism to *effect* delay or queuing *priorities*.

4 Real-Time Distributed Load Balancing

Though the online optimization (16) was separated across time instants, the instantaneous convex problem (25) still requires a centralized solver optimizing the variables of all DCs and MNs jointly, which can be challenging if, e.g., the number of variables is very large. Our next goal is to develop an algorithm that, for each time slot t, finds the optimal solution distributedly across the network entities (MNs and DCs) using only local exchanges. Distributed algorithms exhibit a number of attractive features in networked setups, including low computational complexity, robustness and privacy [26].

Toward these objectives, we will again rely on dual decomposition based approaches. Specifically, we further dualize the constraint (16j) in (25), which couples the optimization variables among MNs and DCs. The fact that the constraint is instantaneous means that it has to be satisfied per time instant t (or equivalently per realization $\mathbf{s}_t$). As a result, the algorithms developed in this section have to run several iterations per time instant (those can be thought of as micro-slots), and the overall network optimization algorithm operates in two timescales.[1] With $\boldsymbol{\pi} := [\pi_1, \ldots, \pi_I]^{\top} \in \mathbb{R}^I$ denoting the instantaneous Lagrange multipliers associated with (16j) in (25), the partial Lagrangian of the instantaneous problem in (25) can be written as [cf. $\Phi(\mathbf{x}_t;\mathbf{s}_t)$ in (25)]

$$\tilde{\mathcal{L}}(\mathbf{x},\boldsymbol{\pi}) := \Phi(\mathbf{x}) + \sum_{i\in\mathcal{I}}\pi_i\left(\sum_{j\in\mathcal{J}} v_{i,j} + \sum_{q\in\mathcal{Q}} w_{i,q} - d_i\right), \tag{26}$$

where the stochastic dual variable $\boldsymbol{\lambda}_t$ is implicit in $\Phi(\mathbf{x})$ since $\boldsymbol{\lambda}_t$ is updated in a slower time scale and can be regarded as constant when solving (25). With $\tilde{\mathcal{X}}$ denoting feasible set defined by the constraints (16d), (16e), (16f), (16g), (16h), and (16i), the Lagrange dual function of (25) is $\tilde{\mathcal{D}}(\boldsymbol{\pi}) := \min_{\mathbf{x}\in\tilde{\mathcal{X}}} \tilde{\mathcal{L}}(\mathbf{x},\boldsymbol{\pi})$, and the dual problem of (25) is

$$\max_{\boldsymbol{\pi}}\ \tilde{\mathcal{D}}(\boldsymbol{\pi}). \tag{27}$$

It is worth stressing that different from the dual formulation in (21), which facilitates the implementation of stochastic approximation schemes, the goal of the dual relaxation in (27) is to obtain a fully distributed algorithm, which implies that the computation and communication tasks can be carried out at each MN and DC.

To this end, we propose two gradient methods for solving (27): a DSA that can be used for any convex formulation, and a dual *accelerated* gradient method that requires some additional assumptions.

4.1 DSA-Based Solution

We consider first DSA, which is the workhorse method to find the optimal Lagrange multipliers [27]. With ℓ denoting the iteration (micro-slot) index, the optimal $\boldsymbol{\pi}^*$ is found upon running

$$\boldsymbol{\pi}(\ell+1) = \boldsymbol{\pi}(\ell) + \beta(\ell)\nabla\tilde{\mathcal{D}}(\boldsymbol{\pi}(\ell)), \tag{28}$$

where $\beta(\ell)$ is the stepsize, and the gradient evaluated at $\boldsymbol{\pi}(\ell)$ is given by

$$\nabla\tilde{\mathcal{D}}(\pi_i(\ell)) = \sum_{j\in\mathcal{J}} v_{i,j}(\ell) + \sum_{q\in\mathcal{Q}} w_{i,q}(\ell) - d_i(\ell),\ \forall i. \tag{29}$$

1. As all problems here are instantaneous (for a certain t and $\mathbf{s}_t$). Time index t will be dropped throughout Section 4 for brevity.

As before, $\mathbf{x}(\ell)$ stands for the minimizer of the Lagrangian $\tilde{\mathcal{L}}(\mathbf{x},\boldsymbol{\pi})$ when $\boldsymbol{\pi}=\boldsymbol{\pi}(\ell)$. Partitioning $\min_{\mathbf{x}\in\tilde{\mathcal{X}}}\tilde{\mathcal{L}}(\mathbf{x},\boldsymbol{\pi})$, per iteration ℓ, each DC i needs to obtain a tentative power allocation $\{d_i(\ell),P_i^{\mathrm{g}}(\ell),P_i^{\mathrm{b}}(\ell)\}$ by solving

$$\min_{d_i,P_i^{\mathrm{g}},P_i^{\mathrm{b}}} \; G_i^{\mathrm{e}}(P_i^{\mathrm{m}})+G_i^{\mathrm{c}}(P_i^{\mathrm{g}})+G_i^{\mathrm{b}}(P_i^{\mathrm{b}})+\lambda_i^{\mathrm{b}}P_i^{\mathrm{b}}-\pi_i(\ell)d_i \quad \text{s.t. (16d)}-\text{(16f)} \tag{30}$$

and a delay-tolerant workload schedule $\{w_{i,q}(\ell)\}$ by solving

$$\min_{w_{i,q}} \Big(\pi_i(\ell)-\lambda_{i,q}^{\mathrm{dc}}\Big)w_{i,q}-U_{i,q}^{\mathrm{w}}(w_{i,q}) \tag{31}$$

while each MN j has to obtain $\{p_j(\ell),v_{i,j}(\ell),\tilde{w}_{i,j,q}(\ell)\}$ via

$$\min_{p_j,v_{i,j},\tilde{w}_{i,j,q}} \sum_{i\in\mathcal{I}_j}\Bigg[\sum_{q\in\mathcal{Q}}\Big(\lambda_{i,q}^{\mathrm{dc}}-\lambda_{j,q}^{\mathrm{mn}}\Big)\tilde{w}_{i,j,q}+\pi_i(\ell)v_{i,j} + G_{i,j}^{\mathrm{d}}\Bigg(v_{i,j}+\sum_{q\in\mathcal{Q}}\tilde{w}_{i,j,q}\Bigg)\Bigg]+\frac{(p_j)^2}{2\kappa_j}-U_j^{\mathrm{v}}(p_j;V_j) \quad \text{s.t.} \quad \text{(16g)}-\text{(16i)}. \tag{32}$$

The DSA enjoys convergence guarantees if a sequence of non-summable diminishing stepsizes is chosen to satisfy $\lim_{\ell\to\infty}\beta(\ell)=0$ and $\sum_{\ell=0}^{\infty}\beta(\ell)=\infty$ [27]. Since (25) is convex, the duality gap is zero, and the minimizer of the Lagrangian yields the optimal solution to the primal problem (25). Alternatively, if a constant stepsize β is adopted, the subgradient iterations (28) are guaranteed to converge to a neighborhood of the optimal $\boldsymbol{\pi}^*$ for the dual problem (27), and the running average of the primal variables will converge to the optimal solution [27]. In practice, the iterations can be stopped once a pre-specified tolerance (or duality gap) is met. It is also worth noting that the DSA is fairly robust and exhibits a number of features that are attractive for networked setups, including the fact of converging to a near-optimal solution even when the information exchanges (e.g., the multipliers) are noisy or sporadically lost. This can happen in the presence of noise in the communication links across the network; see, e.g., [28].

4.2 Accelerated FISTA-Based Solution

Though universally applicable and widely used, DSA does not leverage properties that are specific to the problem at hand, including differentiability of the dual function and Lipschitz continuity of its gradient. In this section, we improve the solver for the problem at the fast timescale (i.e., (25)) by advocating an alternative approach based on FISTA [29], which exhibits faster convergence rate than that of DSA in (28). This is important, because for any t a new instance of (25) needs to solved in real time.

Algorithm 1. Dual FISTA Iteration for (27)

1: **Initialize:** with proper $\boldsymbol{\pi}(0)$, $\boldsymbol{\pi}(1)$, $\theta_\pi(0)$ and stepsize β
2: **for** $\ell=1,2\ldots$**do**
3: Update $\theta_\pi(\ell+1)$ via (33b).
4: Update $\bar{\boldsymbol{\pi}}(\ell)$ via (33a).
5: DCs send $\bar{\boldsymbol{\pi}}(\ell)$ to MNs.
6: Each MN and DC locally solve (30), (31), and (32) using $\boldsymbol{\pi}=\bar{\boldsymbol{\pi}}(\ell)$ to obtain the virtual decision $\mathbf{x}(\ell)$.
7: Update Lagrange multipliers $\boldsymbol{\pi}(\ell)$ via (33c).
8: **end for**

Inheriting Nesterov's acceleration scheme [30], FISTA was originally developed in the context unconstrained primal optimization [29], and has been recently applied to dual problems for network utility maximization [31]. Unlike the dual gradient iteration (28), which relies only on the current iterate, FISTA leverages the memory of one previous iterate aiming to improve convergence. Per iteration (microslot) ℓ, FISTA constructs an intermediate iterate $\bar{\boldsymbol{\pi}}(\ell)$ by using an affine combination of the two most recent iterates $\boldsymbol{\pi}(\ell)$ and $\boldsymbol{\pi}(\ell-1)$, that is

$$\bar{\boldsymbol{\pi}}(\ell)=\left(1-\frac{1-\theta_\pi(\ell-1)}{\theta_\pi(\ell)}\right)\boldsymbol{\pi}(\ell)+\frac{1-\theta_\pi(\ell-1)}{\theta_\pi(\ell)}\boldsymbol{\pi}(\ell-1), \tag{33a}$$

where the weights are "optimally" updated as [29]

$$\theta_\pi(\ell)=\frac{1}{2}\left(1+\sqrt{1+4\theta_\pi^2(\ell-1)}\right). \tag{33b}$$

By computing the dual gradient at the intermediate iterate $\bar{\boldsymbol{\pi}}(\ell)$, the dual variable $\boldsymbol{\pi}(\ell)$ is updated using a gradient ascent step based on $\bar{\boldsymbol{\pi}}(\ell)$, that is

$$\pi_i(\ell)=\bar{\pi}_i(\ell)+\beta\Bigg(\sum_{j\in\mathcal{J}}v_{i,j}(\ell)+\sum_{q\in\mathcal{Q}}w_{i,q}(\ell)-d_i(\ell)\Bigg),\ \forall i, \tag{33c}$$

where β is a proper stepsize. The dual FISTA iteration is summarized in Algorithm 1. Note that the key difference between the DSA iteration (28) and the dual FISTA iteration (33) is the intermediate iterate $\bar{\boldsymbol{\pi}}(\ell)$ along with the gradient at $\bar{\boldsymbol{\pi}}(\ell)$. Intuitively, by combining the previous two dual iterates, the "smoothed" iterates $\bar{\boldsymbol{\pi}}(\ell)$ can mitigate the undesirable oscillation of the simple gradient ascent iteration, thus achieving fast convergence. A more detailed interpretation of Nesterov's acceleration and FISTA can be found in, e.g., [32].

Convergence of FISTA requires: a) the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ to be differentiable; and b) $\nabla\tilde{\mathcal{D}}$ to be Lipschitz continuous. Hence, in the remainder of the section we first elaborate on these two conditions, and then assert the convergence of FISTA formally in Proposition 3.

To satisfy a) in our setup, the following assumption is required: *(as1) for a given t, the network cost $\Psi(\mathbf{x})$ is strongly convex w.r.t.* $\mathbf{x}$. From an engineering perspective, assuming strong convexity is reasonable. Oftentimes in practice the *marginal* rewards (costs) are *monotonically* decreasing (increasing), which does guarantee strong convexity. But even if they are not, one can approximate $\Phi(\mathbf{x})$ in (25) by the regularized strongly convex cost function $\Phi(\mathbf{x})+(\epsilon/2)\|\mathbf{x}\|^2$, with $\epsilon>0$. While the regularizer may introduce a small optimality loss, the solution is feasible and the loss is proportional to ϵ, which is typically selected small (see Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2016.2636210, for details). Indeed, since our ultimate goal for DGLB is to have an $\mathcal{O}(\mu)$-optimal online solution, we will show that it suffices to set $\epsilon=\mathcal{O}(\mu)$. Once (as1) holds, differentiability of the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ follows readily, as formally stated next [33, Lemma 1].

Proposition 1. *For a given $\boldsymbol{\pi}$, the partial Lagrangian (26) has a unique minimizer; thus, the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ is continuously differentiable.*

Proof. See [33, Lemma 1]. □

However, finding the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ required in b) is nontrivial due to the coupling among primal variables. To circumvent this impasse, we introduce an equivalence between differentiability of a convex function and strong convexity of its conjugate to facilitate the derivation of the Lipschitz constant; see [31, Lemma II.1].

Lemma 1. *Let h: $\mathbb{R}^n \to (-\infty, \infty]$ be a proper, lower semicontinuous, convex function, and constant $\sigma > 0$. The following statements are equivalent: S1) Function h is differentiable and its gradient ∇h is Lipschitz continuous with constant $\frac{1}{\sigma}$; S2) The conjugate function h^*: $\mathbb{R}^n \to (-\infty, \infty]$ is σ-strongly convex.*

Leveraging Lemma 1, the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ can be found by analyzing the convexity of the primal objective. The precise result is given in the next proposition.

Proposition 2. *The Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ is*

$$L := (J+Q+1) \times \max_{q,j,i}\left\{\frac{1}{\sigma(U^{\mathrm{w}}_{i,q})}, \frac{1}{\sigma(G^{\mathrm{b}}_{i})}, \frac{1}{\sigma(G^{\mathrm{d}}_{i,j})}, \frac{1}{\sigma(G^{\mathrm{c}}_{i})}, \frac{M_i D_i^2}{2\underline{\alpha}^{\mathrm{s}}_i \varrho}, \frac{2\kappa_j^2}{2\kappa_j + \sigma(U^{\mathrm{v}}_j)}\right\}, \tag{34}$$

where $\sigma(U^{\mathrm{w}}_{i,q})$, $\sigma(G^{\mathrm{b}}_{i})$, $\sigma(G^{\mathrm{d}}_{i,j})$, $\sigma(G^{\mathrm{c}}_{i})$, and $\sigma(U^{\mathrm{v}}_{j})$ are defined in Proposition 6 in Appendix B, available in the online supplemental material.

Proof. See Appendix B, available in the online supplemental material. □

With the definition of L, we are ready to establish the convergence result [30], [31], which closes this section.

Proposition 3. *If L denotes the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ in (34), and the step-size $\beta \in (0, \frac{1}{L}]$, then Algorithm 1 converges to optimal dual variable $\boldsymbol{\pi}^*$. And for $\ell \geq 1$, it satisfies*

$$\tilde{\mathcal{D}}(\boldsymbol{\pi}^*) - \tilde{\mathcal{D}}(\boldsymbol{\pi}(\ell)) \leq \frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2}{\beta(\ell+1)^2}. \tag{35}$$

Proof. See the proof of [29, Theorem 4.4]. □

4.3 Diagonally Weighted FISTA

Computing the Lipschitz constant L, whose value has to be known to set β in (33), requires in general communication among all the DCs and MNs [cf. (34)], which may be difficult (or costly). In this section, we consider the scaled version of FISTA, where each dual variable π_i is updated using a *different stepsize* with limited information exchanges.

Collect the I stepsizes in the $I \times I$ diagonal matrix $\boldsymbol{\Lambda}$ whose ith diagonal element is given by

$$\Lambda_{ii} = \sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}_j}\left\{\frac{1}{\sigma(G^{\mathrm{d}}_{i,j})}, \frac{2\kappa_j^2}{2\kappa_j + \sigma(U^{\mathrm{v}}_j)}\right\} + \sum_{q\in\mathcal{Q}} \frac{1}{\sigma(U^{\mathrm{w}}_{i,q})} + \max\left\{\frac{1}{\sigma(G^{\mathrm{b}}_{i})}, \frac{1}{\sigma(G^{\mathrm{c}}_{i})}, \frac{M_i D_i^2}{2\underline{\alpha}^{\mathrm{s}}_i \varrho}\right\}. \tag{36}$$

Compared to the standard FISTA in Section 4.2, here each DC only needs to know the global information $\sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}_j}\left\{1/\sigma(G^{\mathrm{d}}_{i,j}), 2\kappa_j^2/\left(2\kappa_j + \sigma(U^{\mathrm{v}}_j)\right)\right\}$, which can be obtained from the subset of MNs the particular DC is connected to.

With the diagonal scaling matrix $\boldsymbol{\Lambda}$ defined in (36), we can consequently establish the next proposition.

Proposition 4. *If the update for $\pi_i(\ell)$ in (33c) is replaced with*

$$\pi_i(\ell) = \bar{\pi}_i(\ell) + \Lambda_{ii}^{-1}\left(\sum_{j\in\mathcal{J}} v_{i,j}(\ell) + \sum_{q\in\mathcal{Q}} w_{i,q}(\ell) - d_i(\ell)\right), \quad \forall i$$

then Algorithm 1 converges to the optimal dual variable $\boldsymbol{\pi}^*$. And for $\ell \geq 1$, it holds that

$$\tilde{\mathcal{D}}(\boldsymbol{\pi}^*) - \tilde{\mathcal{D}}(\boldsymbol{\pi}(\ell)) \leq \frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2_{\boldsymbol{\Lambda}}}{(\ell+1)^2}. \tag{37}$$

Proof. See Appendix C, available in the online supplemental material. □

Notice that since $\boldsymbol{\Lambda} \preceq L\mathbf{I}$, it follows that

$$\frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2_{\boldsymbol{\Lambda}}}{(\ell+1)^2} \leq \frac{2L\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2}{(\ell+1)^2} \leq \frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2}{\beta(\ell+1)^2}.$$

This implies that along with the reduction of the communication overhead, the scaled FISTA also enjoys a faster convergence rate.

4.4 Real-Time Distributed Implementation

Integrating the stochastic DSA in Section 3.3 with the dual FISTA algorithm in Section 4.3 gives rise to the DGLB algorithm proposed in this paper, whose steps are described in detail in Algorithm 2.

Algorithm 2. Distributed Geographical Load Balancing

1: **Initialize** Lagrange multipliers $\boldsymbol{\lambda}_0$, and stepsizes $\mu, \boldsymbol{\Lambda}$.
2: **Per slot** t, observe the state $\boldsymbol{s}_t$, obtain $\boldsymbol{\lambda}_t$ from the step 7 at iteration $t-1$, and then run the following tasks.
3: **Signaling exchange**. Obtain $\boldsymbol{\pi}^*_t$ using either the DSA (28) or the FISTA updates (33).
4: **MN pricing and routing**. Each MN solves (32) using $\boldsymbol{\pi}^*_t$, and obtains $\{p_{j,t}, v_{i,j,t}, \tilde{w}_{i,j,q,t}\}$. Offer incentive payment $p_{j,t}$ to the end users nearby, and perform workload routing $\{v_{i,j,t}, \tilde{w}_{i,j,q,t}\}$ based on the actual arrival rates.
5: **DC workload schedule**. Obtain $\{w_{i,q,t}\}$ by solving (31) using $\pi^*_{i,t}$. Process IWs based on $\{v_{i,j,t}\}$, and schedule DWs in each class according to $\{w_{i,q,t}\}$.
6: **DC energy schedule and trading**. Obtain $\{d_{i,t}, P^{\mathrm{g}}_{i,t}, P^{\mathrm{b}}_{i,t}\}$ by solving (30). Perform energy transaction with the main grid; that is, buy the energy amount $[P^{\mathrm{m}}_{i,t}]^+$ with price $\alpha^t_{b,t}$ upon energy deficit, or, sell the energy amount $[P^{\mathrm{m}}_{i,t}]^-$ with price $\alpha^t_{s,t}$ upon energy surplus. Perform battery (dis)charging according to $P^{\mathrm{b}}_{i,t}$, and plan CG generations.
7: **Update the stochastic multipliers** $\boldsymbol{\lambda}_{t+1}$. With $\{P^{\mathrm{b}}_{i,t}, \tilde{w}_{i,j,q,t}, w_{i,q,t}\}$ available, DCs update Lagrange multipliers $\{\lambda^{\mathrm{b}}_{i,t+1}\}$ and $\{\lambda^{\mathrm{dc}}_{i,q,t+1}\}$ via (24a) and (24b), and MNs update Lagrange multipliers $\{\lambda^{\mathrm{mn}}_{j,q,t+1}\}$ via (24c).

Regarding steps 4 and 5, if the real-time energy purchase and selling prices are identical at each DC [34], i.e., $\alpha^{\mathrm{p}}_{i,t}=\alpha^{\mathrm{s}}_{i,t}$, $\forall t$, the DC subproblems (30) and (31) can be solved in closed-form, as formalized next.

Proposition 5. *For DC subproblem (30), the optimal solutions can be expressed in a closed-form*

$$P^{\mathrm{g}}_i(\ell)=\Big[(\nabla G^{\mathrm{g}}_i)^{-1}(\alpha^{\mathrm{p}}_i)\Big]_0^{\overline{P}^{\mathrm{g}}_i},\ P^{\mathrm{b}}_i(\ell)=\Big[(\nabla G^{\mathrm{b}}_i)^{-1}(-\lambda^{\mathrm{b}}_i-\alpha^{\mathrm{p}}_i)\Big]_{\underline{P}^{\mathrm{b}}_i}^{\overline{P}^{\mathrm{b}}_i}$$

and likewise

$$d_i(\ell)=\big[(M_iD_i^2\pi_i(\ell))/(2\varrho(1+e_i)\alpha^{\mathrm{p}}_i)\big]_0^{M_iD_i}.$$

For subproblem (31), the optimal solution is

$$w_{i,q}(\ell)=\Big[(\nabla U^{\mathrm{w}}_{i,q})^{-1}(\pi_i(\ell)-\lambda^{\mathrm{dc}}_{i,q})\Big]_0^{\infty}.$$

Proof. See Appendix D, available in the online supplemental material. □

Notice that the communication overhead of DGLB is fairly low. While DC sub-problems have closed-form solutions, MN sub-problems (32) can be solved in parallel, reducing the per-iteration complexity. In addition, leveraging the accelerated convergence offered by FISTA, the algorithm usually finds the solution within *tens* of iterations.

5 Performance Guarantees

To arrive at our main claim, we begin by quantifying the optimality gap of the proposed DGLB. Based on the results [6], [24], the following lemma holds true.

Lemma 2. *If the random state $\boldsymbol{s}_t$ is either i.i.d. or follows a finite state ergodic Markov chain, and the random duration of the renewal interval of the Markov chain ΔT_n satisfies $\mathbb{E}[\Delta T_n^2]<\infty$, then the limiting time-averaged net-cost under the proposed online algorithm satisfies*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Psi(\mathbf{x}_t;\mathbf{s}_t)]\leq\Psi^*+\mu M\frac{\mathbb{E}[\Delta T_n^2]}{\mathbb{E}[\Delta T_n]},$$

where $\Delta T_n=1$ for the i.i.d. case, the constant M is defined as

$$M:=\frac{1}{2}\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\left(\max\left\{\overline{W}_{j,q},\sum_{i\in\mathcal{I}}B_{j,i}\right\}\right)^2+\frac{1}{2}\sum_{i\in\mathcal{I}}\left(\sum_{q\in\mathcal{Q}}\big(\max\{M_iD_i,\sum_{j\in\mathcal{J}}B_{j,i}\}\right)^2+\big(\max\{\overline{P}^{\mathrm{b}}_i,-\underline{P}^{\mathrm{b}}_i\}\big)^2\Big)$$

and Ψ^* is the optimal value of (16) under any feasible control.

Proof. See Appendix E, available in the online supplemental material. □

Lemma 2 asserts that our DGLB can achieve a near-optimal objective value for (16). However, since DGLB approximates a relaxation of (16) [cf. (18)], the resultant dynamic control policy is not guaranteed to be feasible. In the sequel, we will establish that, when properly initialized, DGLB indeed yields a feasible policy for (16). To achieve this, we start by characterizing a pair of properties of the optimal policy.

Lemma 3. *If $\overline{\alpha}^{\mathrm{p}}_i:=\max\{\alpha^{\mathrm{p}}_{i,t},\forall t\}$ and $\underline{\alpha}^{\mathrm{s}}_i:=\min\{\alpha^{\mathrm{s}}_{i,t},\forall t\}$, the real-time battery (dis)charging decisions $P^{\mathrm{b}}_{i,t}$ generated by the DGBL algorithm satisfy: i) $P^{\mathrm{b}}_{i,t}=\underline{P}^{\mathrm{b}}_i$, if $\lambda^{\mathrm{b}}_{i,t}>-\underline{\alpha}^{\mathrm{s}}_i-\partial\underline{G}^{\mathrm{b}}_i$; and, ii) $P^{\mathrm{b}}_{i,t}=\overline{P}^{\mathrm{b}}_i$, if $\lambda^{\mathrm{b}}_{i,t}<-\overline{\alpha}^{\mathrm{p}}_i-\partial\overline{G}^{\mathrm{b}}_i$.*

Lemma 3 nicely entails the economic interpretation of Lagrange multipliers. Indeed, the stochastic multiplier $\lambda^{\mathrm{b}}_{i,t}$ can be viewed as the instantaneous charging price. When the price $\lambda^{\mathrm{b}}_{i,t}$ is high enough, the optimal decision is to fully discharge the battery; and when the price $\lambda^{\mathrm{b}}_{i,t}$ is sufficiently low, it will be optimal to charge the battery as much as possible. Based on this salient optimal structure, Lemma 3 allows us to further establish the following result.

Lemma 4. *If the stepsize satisfies $\mu\geq\underline{\mu}$, where*

$$\underline{\mu}:=\max_i\Big\{\big(\overline{\alpha}^{\mathrm{p}}_i+\partial\overline{G}^{\mathrm{b}}_i-\underline{\alpha}^{\mathrm{s}}_i-\partial\underline{G}^{\mathrm{b}}_i\big)/\big(\overline{Y}^{\mathrm{b}}_i-\underline{Y}^{\mathrm{b}}_i+\underline{P}^{\mathrm{b}}_i-\overline{P}^{\mathrm{b}}_i\big)\Big\},$$

then the stochastic multipliers generated by DGLB satisfy $-\overline{\alpha}^{\mathrm{p}}_i-\partial\overline{G}^{\mathrm{b}}_i+\mu\underline{P}^{\mathrm{b}}_i\leq\lambda^{\mathrm{b}}_{i,t}\leq\mu\overline{Y}^{\mathrm{b}}_i-\mu\underline{Y}^{\mathrm{b}}_i-\overline{\alpha}^{\mathrm{p}}_i-\partial\overline{G}^{\mathrm{b}}_i+\mu\underline{P}^{\mathrm{b}}_i,\ \forall i,t$.

These two lemmas are generalizations of [8, Lemma 4] and [8, Lemma 5]; their proof is omitted here for brevity.

Consider now the linear mapping

$$Y^{\mathrm{b}}_{i,t}=\big(\lambda^{\mathrm{b}}_{i,t}+\overline{\alpha}^{\mathrm{p}}_i+\partial\overline{G}^{\mathrm{b}}_i\big)/\mu+\underline{Y}^{\mathrm{b}}_i-\underline{P}^{\mathrm{b}}_i,\quad\forall i.\tag{38}$$

It can be readily seen from Lemma 4 that $\underline{Y}^{\mathrm{b}}_i\leq Y^{\mathrm{b}}_{i,t}\leq\overline{Y}^{\mathrm{b}}_i$ holds for all i and t; i.e., (16c) are always satisfied under the proposed online scheme. With the battery (dis)charging dynamics (16b) naturally performed, feasibility of the control actions $\mathbf{x}(\lambda_t)$ can be maintained for the original problem, provided that we select a stepsize $\mu\geq\underline{\mu}$.

Using Lemmas 2 and 4, the following theorem assessing the feasibility and optimality of DGLB, which is the main result of this section, can be established.

Theorem 1. *Upon setting $\lambda^{\mathrm{b}}_{i,0}=\mu Y^{\mathrm{b}}_{i,0}-\mu\underline{Y}^{\mathrm{b}}_i-\overline{\alpha}^{\mathrm{p}}_i-\partial\overline{G}^{\mathrm{b}}_i+\mu\underline{P}^{\mathrm{b}}_i$, $\forall i$, and selecting a stepsize $\mu\geq\underline{\mu}$, the DGLB algorithm yields a feasible dynamic control scheme for (16), which satisfies*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Psi(\mathbf{x}_t;\mathbf{s}_t)]\leq\Psi^*+\mu M\frac{\mathbb{E}[\Delta T_n^2]}{\mathbb{E}[\Delta T_n]},$$

where ΔT_n, M and $\underline{\mu}$ are specified in Lemmas 2, 3 and 4.

Proof. Theorem follows from Lemmas 2 and 4 readily. □

The theorem asserts that with a proper initialization and $\mu\geq\underline{\mu}$, DGLB is feasible for the original problem in (16), and incurs a bounded optimality loss. Since this loss is increasing with μ, setting $\mu=\underline{\mu}$ minimizes the gap and limits the performance loss to $\underline{\mu}M\mathbb{E}[\Delta T_n^2]/\mathbb{E}[\Delta T_n]$. Scenarios where both the difference between purchase and selling prices $(\overline{\alpha}^{\mathrm{p}}_i-\underline{\alpha}^{\mathrm{s}}_i)$ and the difference between marginal charging and discharging costs $\partial\overline{G}^{\mathrm{b}}_i-\partial\underline{G}^{\mathrm{b}}_i$ approach zero allow for selecting $\underline{\mu}$ very small [cf. Lemma 4], so that the optimality loss is practically null. The same is true in scenarios where the battery capacities $\overline{Y}^{\mathrm{b}}_i$ are very large. This makes sense intuitively because as both $(\overline{\alpha}^{\mathrm{p}}_i-\underline{\alpha}^{\mathrm{s}}_i)$ and $\partial\overline{G}^{\mathrm{b}}_i-\partial\underline{G}^{\mathrm{b}}_i$ approach zero, purchasing extra energy to charge the batteries (if they are close to empty) or

TABLE 2
DC Power-Related Parameters

$\overline{P}_i^{\mathrm{g}}$	$\underline{Y}_i^{\mathrm{b}}$	$\overline{Y}_i^{\mathrm{b}}$	$Y_{i,0}^{\mathrm{b}}$	$\underline{P}_i^{\mathrm{b}}$	$\overline{P}_i^{\mathrm{b}}$	$\overline{P}_i^{\mathrm{s}}$
100	5	100	5	-20	20	1

The units are kW or kWh.

selling it to discharge them (if they are close to full) is always profitable. Similarly, when batteries have large capacity, the upper bounds in (16c) do not hold as equalities, and stationary policies obeying the long-term energy conservation constraint are optimal. Note finally that selecting $\mu > \underline{\mu}$ can be used to reach the close-to-optimal (steady-state) operation point more quickly, but the incurred optimality loss will be higher [8], [24].

Remark 2. While feasibility of the battery dynamics holds for arbitrary sample paths of $\{\boldsymbol{s}_t\}$, near optimality of DGLB is guaranteed under the assumption that the random state $\boldsymbol{s}_t$ is either i.i.d. or follows an ergodic Markov chain. Markovianity is widely used in wireless networks and power system applications to model the stochastic demand, renewable generation, and price processes [35]. Numerical results will further demonstrate that the DGLB can obtain a desirable performance even in real data scenarios.

Remark 3. Readers familiar with the so-called Lyapunov-optimization framework can recognize similarities between the stochastic DSA proposed here, and the tools in [6], [22]. The differences between them can be summarized as follows:

- D1) The Lyapunov-optimization solver relies on the so-called "virtual queues" to ensure that long-term average constraints are met, where the tuning parameter V in [6], [22] corresponds to the inverse of the stepsize μ in the stochastic DSA. In contrast, "virtual queues" emerge naturally as Lagrange multiplier iterates in our stochastic DSA;
- D2) Leveraging duality and stochastic approximation techniques, the multiplier update in the stochastic DSA is also easy to interpret. The multipliers for instance, can be viewed as the instantaneous charging prices, revealing the intuition behind workload routing, scheduling and real-time (dis)charging decisions, as discussed after Lemma 4; and
- D3) Results from duality theory, including sensitivity and weak duality, can be used to characterize the performance of our stochastic DSA (cf. Lemma 3).

6 Numerical Tests

This section presents numerical tests to confirm the analytical claims in Section 5, and demonstrate the merits of the proposed approach. We start by describing the simulation setup. The network considered has $I = 4$ DCs and $J = 4$ MNs located in the eastern, central, mountain and western parts of the US. The number of servers at each DC is $\{M_i\} = \{1{,}000, 750, 750, 1{,}000\}$. One unit of workload is assumed to require the computing resources of five servers ($D_i = 0.2, \forall i$), and the IW curtailment ratio is $\eta = 0.2$. For simplicity, the cooling coefficients at each DC are considered time-invariant with values $\{e_{i,t}\} = \{0.2, 0.3, 0.4, 0.5\}$. The bandwidth limits $\{B_{j,i}\}$ are generated from a uniform distribution within $[20, 300]$, and the communication cost is $G_{i,j}^{\mathrm{d}}(\{\tilde{w}_{i,j,q,t}\}_{q\in\mathcal{Q}}, v_{i,j,t}) = c_{i,j}^{\mathrm{d}}(v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t})^2 + v_{i,j,t}^2 + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t}^2$, with $c_{i,j}^{\mathrm{d}}$ inversely proportional to $B_{j,i}$. We consider $Q = 2$ types of DW tasks, and the revenue for each type is the same at all DCs $U_{i,q}^{\mathrm{w}}(w_{i,q,t}) = -u_q(w_{i,q,t})^2 + 50u_q w_{i,q,t}$ with u_q being uniformly distributed within $[1, 3]$ cents/(unit)^2. The coefficient κ_j in $G_j^{\mathrm{u}}(\check{V}_{j,t})$ is generated from a uniform distribution within $[1, 3]$ cents/(unit)^2. Each DC is connected to a microgrid consisting of a CG, an RG, a battery, and facilities for two-way trading with the external market. The power-related parameters are set identically across all DCs as listed in Table 2. The CG cost is $G_i^{\mathrm{c}}(P_{i,t}^{\mathrm{g}}) = c^{\mathrm{g}}(P_{i,t}^{\mathrm{g}})^2 + 10c^{\mathrm{g}}P_{i,t}^{\mathrm{g}}$ with $c^{\mathrm{g}} = 0.5$ cents/(kWh)^2, while the battery (dis)charging cost is $G_i^{\mathrm{b}}(P_{i,t}^{\mathrm{b}}) = c^{\mathrm{b}}(P_{i,t}^{\mathrm{b}})^2$ with $c^{\mathrm{b}} = 1$ cents/(kWh)^2. In addition, the energy purchase price is set equal to the selling price; i.e., $\alpha_{i,t}^{\mathrm{p}} = \alpha_{i,t}^{\mathrm{s}}$, $\forall i, t$, and the duration of a scheduling period (time slot) is one hour.

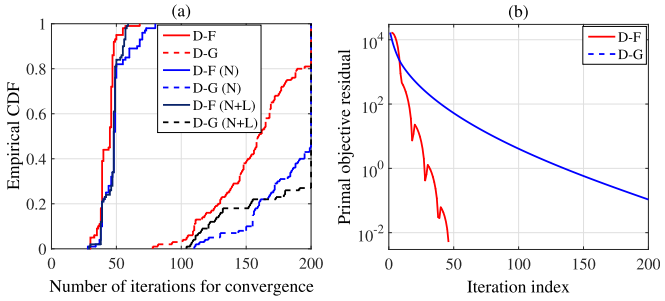

Fig. 2. The left panel shows the empirical CDF of the number of iterations needed to converge ($T = 100$ realizations are considered). For one of those realizations, the right panel plots the evolution of the primal objective residual for the noise-free case. In the legend, "D" denotes distributed, "F" FISTA, "G" gradient, "N" the presence of *noise*, and "L" that some of the multipliers are *lost*.

Two sets of numerical results are presented: one to demonstrate convergence and optimality in synthetic scenarios where the random variables are drawn either from a given distribution, or, an ergodic Markov chain (Section 6.1); and the other one to illustrate performance in a practical scenario using real data (Section 6.2). Note that workload arrivals, energy prices, and renewable generations in Case study 2 are highly correlated over time, so it also serves to assess the applicability of DGLB to non-stationary setups. To benchmark performance of the proposed algorithm, three baseline schemes are tested including both the local load balance (LLB) as well as the geographical load balancing schemes (GLB).

1) *ALG 1 (LLB, with incentive payment, DW scheduling, RES and storages):* ALG 1 is similar to the algorithms in [6], [8], where MNs only route workloads to the closest DC.
2 *ALG 2 (GLB, without DW scheduling, nor incentive payment, with RES and storages):* ALG 2 is similar to the method in [14], that is widely used in practical network systems, where no incentive pricing is used, and all the DWs are processed once they arrive, without any delay.
2 *ALG 3 (GLB, without RES and storages, with DW scheduling and incentive payment):* ALG 3 mimics the algorithm in [13], but with only one-timescale

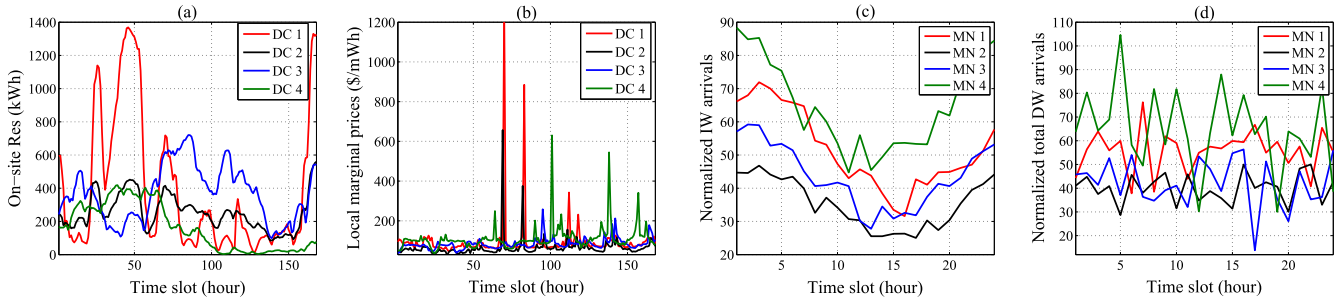

Fig. 3. Time variation of the RES generation, local marginal prices, and IW and DW arrivals used in Test Case 2 [14], [36], [37], [38], [39].

operation, where DCs are only powered by CG or power from the spot market, without considering RES and storage units.

6.1 Case Study 1: Convergence and Optimality

We start by running an experiment where $\boldsymbol{s}_t$ is i.i.d. Specifically, the purchase price $\alpha^{\mathrm{p}}_{i,t}$ is uniformly distributed within $[10, 30]$ \$/kWh, samples of renewables $\{P^{\mathrm{r}}_{i,t}\}$ are generated from a uniform distribution within $[1, 300]$ kWh, IWs $\{V_{j,t}\}$ and class-q DWs $\{W_{j,q,t}\}$ arrive at each MN according to a Poisson process, all with average arrival rates 50 units/slot. The convergence results when solving the real-time problems in (25) are compared in Fig. 2, where $T = 100$ slots are considered, each consisting of up to 200 micro-slots. A sequence of stepsizes $\beta(\ell) = 0.5/\sqrt{\ell}$, $\ell = 1, \ldots, 200$ are employed for the subgradient iteration in (28), the diagonal-scaled stepsize (36) is used for the dual FISTA iteration, and the stopping criteria is either the primal objective residual being smaller than 0.01, or, the number of iterations being greater than 200. In the noise-free case (red lines in Fig. 2a), the dual FISTA (D-F) converges within 60 iterations in all slots, while the dual gradient (D-G) needs more than 150 iterations on average, and fails to converge within 200 iterations in some cases. The accelerated convergence of D-F is further illustrated in Fig. 2b, where the residual reduction per update (micro-slot) is considerably larger for D-F. Regarding robustness, the blue and black lines in Fig. 2a represent the empirical CDFs of iteration complexity when the exchanged multipliers are noisy (under zero-mean Gaussian noise with variance $\sigma^2 = 1$) or sporadically lost (under link outage probability[2] 0.2), respectively. The conclusion is that the number of iterations required for D-G to converge ranges from 70 to 200, while in most cases D-F converges within as few as 45-55 iterations.

The second experiment models $\{\boldsymbol{s}_t\}$ as a Markov chain, where $\alpha^{\mathrm{p}}_{i,t}$ takes values from a three-state set $\{10, 20, 30\}$, $P^{\mathrm{r}}_{i,t}$ from $\{10, 150, 300\}$, and $V_{j,t}$ as well as $W_{j,q,t}$ are drawn from $\{25, 50, 75\}$. The transition matrix of this Markov chain is

$$\mathbf{P} = \begin{pmatrix} 0.2 & 0.3 & 0.5 \\ 0.3 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.4 \end{pmatrix}.$$

Under this setting, the optimality loss of DGLB relative to the optimal solution of the relaxed problem (18) is depicted in Fig. 4 for different stepsizes. The results confirm that the optimality gap vanishes as the stepsize decreases, and also illustrates that larger stepsizes give rise to faster convergence. This nicely matches the theoretical characterization of the optimality loss provided in Theorem 1.

6.2 Case Study 2: Scenario with Real Data

In this test case, the purchase prices $\{\alpha^{\mathrm{p}}_{i,t}\}$ at DCs 1 and $3-4$ are re-scaled from the local real-time data in PJM (eastern), MISO (central), and CAISO (western) during Oct. 01-25, 2015, while the renewable generations $\{P^{\mathrm{r}}_{i,t}\}$ are based on the data during Oct. 01-25, 2012 [36], [37], [38]. Real-time prices and renewable generation in the mountain region are hard to obtain, so that we generate them by averaging and re-scaling the data from central and western areas. As workload traces are not available from public sources, IWs are generated by duplicating the Wikipedia load trace over a 24-hour period [14], while DWs are generated by repeating the hourly MapReduce trace over a day [39]. In both cases, white Gaussian noise with variance randomly drawn between 3–5 dB was added to the original values, which were also re-scaled to model regional differences. The Western Time Zone (UTC-8) was used for time-keeping, and all data was shifted to show the effect of time zone differences. To facilitate the interpretation of the results, the values of $\{\alpha^{\mathrm{p}}_{i,t}\}$ and $\{P^{\mathrm{r}}_{i,t}\}$ over a week are shown in Figs. 3a and 3b, those of IWs and DWs over a day are shown in Figs. 3c and 3d, and their average is listed in Table 3.

Fig. 5 depicts the running average of the network cost (primal objective) of DGLB and ALGs 1-3. Over $T = 600$

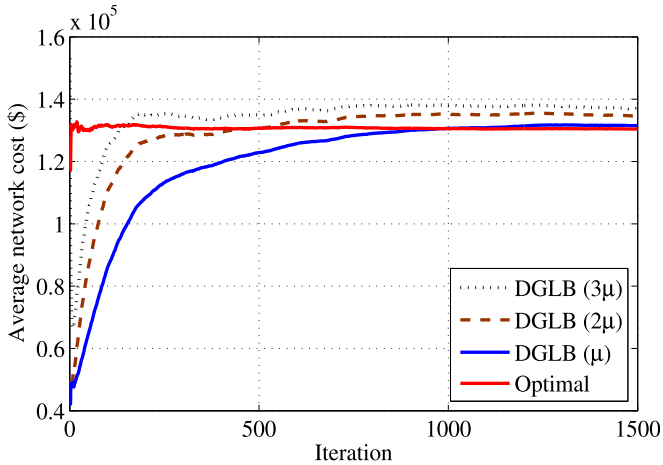

Fig. 4. Optimality gap of DGLB for different stepsizes ($\mu = 0.35$).

2. In the case of a link outage, the subproblems (30), (31), and (32) at iteration ℓ are solved using the outdated Lagrange multipliers $\boldsymbol{\pi}_t(\ell - 1)$.

TABLE 3
Averages of the Time Series used to Run Test Case 2

Index for DC i or MN j	1	2	3	4
Mean ($\alpha^{\mathrm{p}}_{i,t}$), cent/kWh	9.77	6.47	8.32	12.15
Mean ($P^{\mathrm{r}}_{i,t}$), kWh	484.93	290.78	405.54	189.06
Mean ($V_{j,t}$), unit	51.81	34.69	43.32	65.10
Mean ($W_{j,1,t}$), unit	26.68	19.47	19.50	31.37
Mean ($W_{j,2,t}$), unit	29.10	21.26	21.29	34.2064

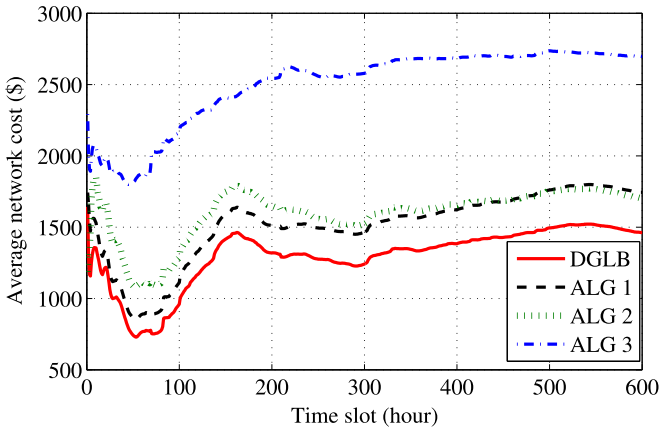

Fig. 5. Comparison of time-average network costs in the DC network.

slots, the average network cost of DGLB is 17 percent lower than that of ALGs 1-2, and around 46 percent lower than that of ALG 3. Recall that ALGs 1-2 are vulnerable to high fluctuations of prices, RES, and workload demands due to the lack of geographical allocation capabilities, or, "workload smoothing" tools (e.g., the incentive payment, the workload delay), and ALG 3 is sensitive to the energy prices since neither RES nor storage units are accounted for. As corroborated by Fig. 5, ALGs 1-3 incur a higher cost since they have to buy more (expensive) energy from the spot market on the peaks of user demand. In contrast, DGLB takes advantage of incentive payments, workload queues as well as RES and storage devices, so it can smooth the workload curvatures and leverage RES and stored energy to avoid future purchases at high prices.

The average energy cost and the ratio of RES to the total energy consumption are shown in Fig. 6. While DGLB generally incurs lower energy cost, ALG 2 (LLB) has the

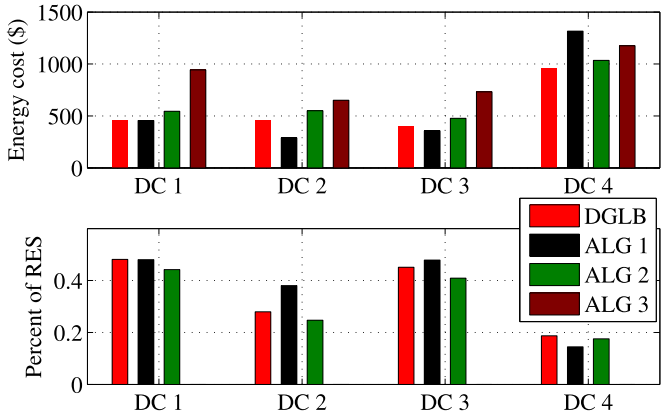

Fig. 6. Comparison of energy cost and RES usage at each DC. The RES usage is the ratio of consumed RES to the total energy consumption.

smallest energy cost and the largest RES usage in DCs 2-3. This makes sense because the workload demands in DCs 2-3 are relatively low [cf. Figs. 3c and 3d]. As GLB schemes smooth the load profile by allocating remote loads to MNs 2-3, LLB only uses local resources, leading to a higher cost and lower RES utilization at DC 4.

To better understand the role of Lagrange multipliers in workload and power balancing, the trajectories of multipliers associated with MN 1 and DC 1 are depicted in Figs. 7a and 7b. The (negative) Lagrange multiplier $-\lambda^{\mathrm{b}}_{1,t}$ in the central panel of Fig. 7a serves as the stochastic discharging price, in the sense that it always increases when the spot market price increases; the Lagrange multiplier $\lambda^{\mathrm{b}}_{1,t}$ precisely maps the evolution of the battery level $C_{1,t}$ corroborating the affine mapping in (38); and the battery, to mitigate the variability of RES, will always discharge when the price $\alpha^{\mathrm{p}}_{1,t}$ is very high. The Lagrange multipliers reflecting workload queue lengths in Fig. 7b are related to the service delay (the Little's law), or, the congestion price at each MN or DC. Per Fig. 7b, the multipliers for DGLB and GLB-based ALG 3 follow almost the same trajectory. Differently, ALG 1 exhibits larger delay, especially at the MN side. Intuitively, this is because in ALG 1, MNs allocate all their workloads to the nearest DC, incurring higher delay when the instantaneous workload arrival rate is very high, or, when the nearest DC is overloaded. Although not shown in

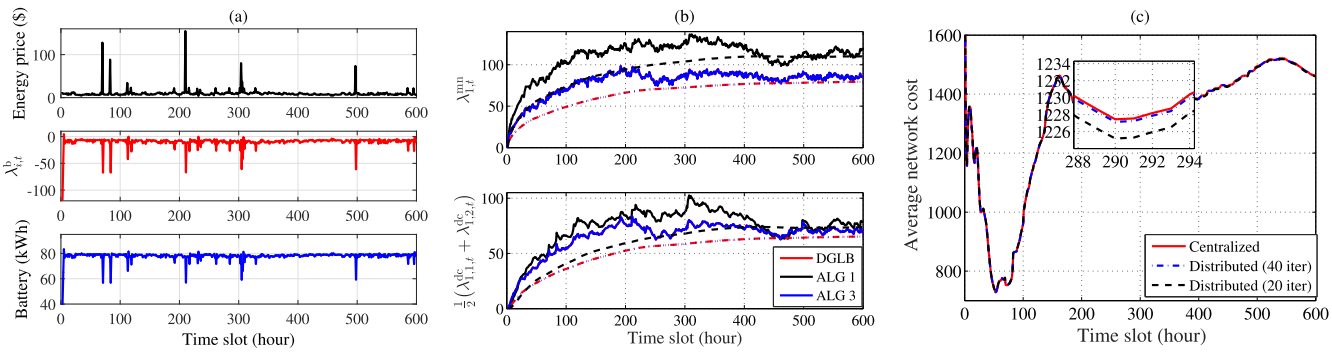

Fig. 7. The left panel shows the evolutions of the price $\alpha^{\mathrm{p}}_{1,t}$, the battery level $C_{1,t}$, and the Lagrange multiplier $\lambda^{\mathrm{b}}_{1,t}$. The middle panel plots the evolutions of Lagrange multipliers $\lambda^{mn}_{1,t}$ and $\left(\lambda^{\mathrm{dc}}_{1,1,t}+\lambda^{\mathrm{dc}}_{1,2,t}\right)/2$ for DGLB. The dashed lines are the running average of instantaneous multipliers. The right panel compares the evolutions of network costs for DGLB using CVX [40] to centrally solve (25), and for DGLB using the (distributed) diagonally weighted FISTA running 20 and 40 iterations.

Fig. 7b, ALG 2 indeed experiences a close-to-zero delay, at the expense of high network cost [cf. Fig. 5].

Since the stopping criteria of FISTA requires a-priori knowledge of the optimal value of (25) that is not available to DGLB in practice, we finally assess the performance of DGLB when running a *fixed* number of iterations in Fig. 7c. Interestingly, the performance of DGLB with 40 iterations is very close to that of the centralized solver (10^{-4} relative optimality loss), and the performance of DGLB running 20 iterations is also good enough in practice (10^{-3} relative optimality loss). As a secondary comment, note that Fig. 7c demonstrates that the distributed DGLB with 20 iterations leads to a slightly lower cost than its more complex counterparts. The reason is that for a very small number of iterations, the constraints (16j) are marginally violated, resulting in a better objective value. In any case, the results in this experiment (together with those in Fig. 2) corroborate the merits of DGLB, and its suitability for distributed real-time implementation.

7 Conclusion

Optimal schemes were designed in this paper for real-time geographical load balancing tailored to the forthcoming sustainable cloud networks. Accounting for the spatio-temporal variability of workloads, renewables, and electricity prices, a stochastic optimization problem was formulated to minimize the long-term aggregate cost of the MN-to-DC network. Leveraging the celebrated dual decomposition methodology, the task was decomposed across time and space, enabling an online distributed implementation. Using a two-timescale approach, a stochastic dual gradient scheme was first implemented to handle the long-term constraints and decouple the optimization across time slots. Then, for each time slot, an accelerated dual gradient method was adopted to tackle the short-term constraints coupling the optimization across DCs. It was analytically established that by properly choosing constant stepsizes and initializations, the novel schemes attain near-optimal performance while respecting the battery capacity and remaining operating constraints, even without knowing the distributions of the underlying stochastic processes. Numerical tests using both synthetic and real data corroborated the effectiveness and merits of the novel approaches.

Acknowledgments

Work in this paper was supported by US National Science Foundation grants 1509040, 1508993, 1423316, 1442686, and by Spanish MINECO Grant TEC2013-41604-R and CAM Grant S2013/ICE-2933.

References

[1] L. A. Barroso, J. Clidaras, and U. Hólzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures Comput. Archit.*, vol. 8, no. 3, pp. 1–154, 2013.

[2] Apple's two massive European data centers. (2015). [Online]. Available: http://www.datacenterknowledge.com/archives/2015/02/23/

[3] Active Power, "Data center thermal runaway. A review of cooling challenges in high density mission critical environments," *White Paper*, 2007. [Online]. Available: www.edsenerji.com.tr/dokuman_indir/16/

[4] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 2007–2015.

[5] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[6] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Modeling Comput. Syst.*, Jun. 2011, pp. 221–232.

[7] Z. Liu, et al., "Renewable and cooling aware workload management for sustainable data centers," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Modeling Comput. Syst.*, Jun. 2012, vol. 40, no. 1, pp. 175–186.

[8] T. Chen, X. Wang, and G. B. Giannakis, "Cooling-aware energy and workload management in data centers via stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 402–415, Mar. 2016.

[9] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, and M. Martonosi, "Capping the brown energy consumption of internet services at low cost," in *Proc. Green Comput. Conf.*, Aug. 2010, pp. 3–14.

[10] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam, "Carbon-aware energy capacity planning for datacenters," in *Proc. IEEE 20th Int. Symp. Modeling Anal. Simul. Comput. Telecommun. Syst.*, Aug. 2012, pp. 391–400.

[11] S. Ren, Y. He, and F. Xu, "Provably-efficient job scheduling for energy and fairness in geographically distributed data centers," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst.*, Jun. 2012, pp. 22–31.

[12] Y. Guo, Y. Gong, Y. Fang, P. P. Khargonekar, and X. Geng, "Energy and network aware workload management for sustainable data centers with thermal storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2030–2042, Aug. 2014.

[13] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1431–1439.

[14] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 854–862.

[15] H. Xu and B. Li, "Reducing electricity demand charge for data centers with partial execution," in *Proc. Int. Conf. Future Energy Syst.*, Jun. 2014, pp. 51–61.

[16] L. Zhang, S. Ren, C. Wu, and Z. Li, "A truthful incentive mechanism for emergency demand response in colocation data centers," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 2632–2640.

[17] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing data center demand response," in *Proc. ACM Int. Conf. Meas. Modeling Comput. Syst.*, Jun. 2014, pp. 111–123.

[18] A. G. Marques, N. Gatsis, and G. B. Giannakis, "Optimal cross-layer design of wireless fading multi-hop networks," in, *Cross Layer Designs in WLAN Systems*. Leicester, U.K.: Troubador Publishing, 2011.

[19] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. Hoboken, NJ, USA: Wiley, 2012.

[20] P. Ramadass, B. Haran, R. White, and B. N. Popov, "Mathematical modeling of the capacity fade of Li-ion cells," *Elsevier J. Power Sources*, vol. 123, no. 2, pp. 230–240, Sep. 2003.

[21] Y. He, S. Elnikety, J. Larus, and C. Yan, "Zeta: Scheduling interactive services with partial execution," in *Proc. ACM Symp. Cloud Comput.*, Oct. 2012, Art. no. 12.

[22] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[23] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[24] A. G. Marques, L. M. Lopez-Ramos, G. B. Giannakis, J. Ramos, and A. J. Caamaño, "Optimal cross-layer resource allocation in cellular networks using channel-and queue-state information," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2789–2807, Jul. 2012.

[25] N. Gatsis and A. G. Marques, "A stochastic approximation approach to load shedding in power networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2014, pp. 6464–6468.

[26] S. Bera, S. Misra, and J. J. Rodrigues, "Cloud computing applications for smart grid: A survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1477–1494, May 2015.

[27] D. P. Bertsekas, *Convex Optimization Theory*. Belmont, MA, USA: Athena Sci., 2009.

[28] N. Gatsis and G. B. Giannakis, "Residential load control: Distributed scheduling and convergence with lost AMI messages," *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 770–786, Jun. 2012.

[29] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
[30] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Soviet Math. Doklady*, vol. 27, no. 2, pp. 372–376, 1983.
[31] A. Beck, A. Nedic, A. Ozdaglar, and M. Teboulle, "An $\mathcal{O}(1/k)$ gradient method for network resource allocation problems," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 64–73, Mar. 2014.
[32] Z. Allen-Zhu and L. Orecchia, "Linear coupling: An ultimate unification of gradient and mirror descent," *arXiv:1407.1537*, Jan. 2015. [Online]. Available: http://arxiv.org/pdf/1407.1537
[33] S. H. Low and D. E. Lapsley, "Optimization flow control-I: Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.
[34] *Energy Primer: A Handbook of Energy Market Basics*, Federal Energy Regulatory Commission, Washington, DC, 2015.
[35] A. M. González, A. M. S. Roque, and J. García-González, "Modeling and forecasting electricity prices with input/output hidden Markov models," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 13–24, Feb. 2005.
[36] MISO market data. (2015). [Online]. Available: https://www.midwestiso.org/MarketsOperations/RealTimeMarketData/
[37] CAISO hourly renewables watch. (2015). [Online]. Available: http://www.caiso.com/green/renewableswatch.html
[38] Hourly electric supply charges in New York, Jan. 2015. [Online]. Available: https://www.nationalgridus.com/
[39] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating MapReduce performance using workload suites," in *Proc. IEEE 19th Annu. Int. Symp. Modelling Anal. Simul. Comput. Telecommun. Syst.*, Jul. 2011, pp. 390–399.
[40] CVX: Matlab software for disciplined convex programming, version 2.1, Sep. 2012. [Online]. Available: http://cvxr.com/cvx
[41] J. Koshal, A. Nedic, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM J. Optimization*, vol. 21, no. 3, pp. 1046–1081, Jul. 2011.
[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.
[43] S. Asmussen, *Applied Probability and Queues*, vol. 51. Berlin, Germany: Springer, 2008.

Tianyi Chen (S'14) received the BEng degree (with highest honors) in communication science and engineering from Fudan University, in 2014, and the MSc degree in electrical and computer engineering from the University of Minnesota (UMN), in 2016, respectively. Since July 2016, he has been working toward the PhD degree at UMN. His research interests lie in online convex optimization, data-driven network optimization with applications to smart grids, sustainable cloud networks, and green communications. He received the Student Travel Grant from the IEEE Communications Society in 2013, a National Scholarship from China in 2013, and the UMN ECE Department Fellowship in 2014. He is a student member of the IEEE.

Antonio G. Marques (SM'13) received the Telecommunications Engineering and Doctorate degree, both with highest honors, from the Carlos III University of Madrid, Spain, in 2002 and 2007, respectively. In 2007, he became a faculty in the Department of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain, where he currently develops his research and teaching activities as an associate professor. From 2005 to 2015, he held different visiting positions with the University of Minnesota, Minneapolis. In 2015 and 2016, he was a visiting scholar with the University of Pennsylvania. His research interests lie in the areas of communication theory, signal processing, and networking. His current research focuses on stochastic resource allocation for wireless networks and smart grids, nonlinear network optimization, and signal processing for graphs. He has served the IEEE in a number of posts (currently, he is an associate editor of the *IEEE Signal Processing Letters*), and his work has been awarded in several conferences and workshops. He is a senior member of the IEEE.

Georgios B. Giannakis (F'97) received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1981. He received the MSc degree in electrical engineering, MSc degree in mathematics, and the PhD degree in electrical engineering from the University of Southern California (USC), in 1983, 1986, and 1986, respectively. From 1982 to 1986, he was with the USC. He was with the University of Virginia from 1987 to 1998, and since 1999 he has been a professor with the University of Minnesota, where he holds an endowed chair in Wireless Telecommunications, a University of Minnesota McKnight presidential chair in ECE, and serves as a director of the Digital Technology Center. His general interests span the areas of communications, networking and statistical signal processing—subjects on which he has published more than 400 journal papers, 680 conference papers, 25 book chapters, two edited books and two research monographs (h-index 119). Current research focuses on learning from big data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-) inventor of 28 patents issued, and the (co-) recipient of eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a fellow of the EURASIP, and has served the IEEE in a number of posts, including that of a distinguished lecturer for the IEEE-SP Society. He is a fellow of the IEEE.