Resource Allocation for Green Cloud Networks under Uncertainty: Stochastic, Robust and Big Data-driven Approaches

---

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Tianyi Chen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Professor Georgios B. Giannakis, Advisor

Professor Nikos Sidiropoulos

Professor Zizhuo Wang

September 2016

Acknowledgments

First and foremost, I would like to express my deepest gratitude and respect to my advisor Prof. Georgios B. Giannakis for his invaluable guidance. Thanks to his incisive foresight and suggestion, I was devoted to working on the areas of stochastic optimization with focus on renewable energy, which constitutes the main threads of this dissertation. I am also grateful for his help on improving my presentation and writing skills. More important than that, I thank him for offering his friendship and encouraging my fruitful collaborations with experts in this field.

Special thanks go to Professors Nikos Sidiropoulos, and Zizhuo Wang for serving on my M.Sc. Thesis Committee, as well as for their patience on coordinating the date of final defense. They are two professors who I met at the very beginning of my graduate studies in UMN, and are really the ones I am looking for to support me and provide valuable suggestions on my research.

The work in this dissertation would not have been possible without the help of the professors and colleagues I have collaborated with during the last two years. Special thanks are due to my undergraduate advisor Prof. Xin Wang, who encouraged me, from a student point of view, to pursue a PhD program in this country and specifically in SPiNCOM, who spent hundreds of hours discussing with me about my "innocent ideas" and correcting my rough drafts. I also wish to give credit and express my warmest thanks to Prof. Antonio Marques, Prof. Vassilis Kekatos, Prof. Alejandro Ribeiro, Prof. Longbo Huang, Prof. Qing Ling, Dr. Yu Zhang, Dr. Aryan Mokhtari, and all other SPiNCOMers.

Last but not least, I would also like to thank my family for all their unconditional support along the way. Particularly, I thank my mother for educating me to be energetic, determinant, ambitious, and persistent in pursuing those dreams deeply rooted in my soul. As a closing comment, I want to remind myself that: I have a dream; always believe in the power of having a dream; always trust the power of consistently pursuing one's dream; and never doubt that something wonderful is about to happen.

Abstract

Major improvements have propelled the development of worldwide Internet systems during the past decade. To meet the growing demand in massive data processing, a large number of geographically-distributed data centers begin to surge in the era of data deluge and information explosion. Along with their remarkable expansion, contemporary cloud networks are being challenged by the growing concerns about global warming, due to their substantial energy consumption. Hence, the infrastructure of future data centers must be energy-efficient and sustainable. Fortunately, supporting technologies of *smart grids*, *big data analytics* and *machine learning*, are also developing rapidly. These considerations motivate well the present thesis, which mainly focuses on developing interdisciplinary approaches to offer sustainable resource allocation for future cloud networks, by leveraging three intertwining research subjects.

The modern smart grid has many new features and advanced capabilities including e.g., high penetration of renewable energy sources, and dynamic pricing based demand-side management. Clearly, by integrating these features into the cloud network infrastructure, it becomes feasible to realize its desiderata of reliability, energy-efficiency and sustainability. Yet, full benefits of the renewable energy (e.g., wind and solar) can only be harnessed by properly mitigating its intrinsically stochastic nature, which is still a challenging task. This prompts leveraging the huge volume of historical data to reduce the stochasticity of online decision making. Specifically, valuable insights from big data analytics can enable a markedly improved resource allocation policy by learning historical user and environmental patterns. Relevant machine learning approaches can further uncover "hidden insights" from historical relationships and trends in massive datasets.

Targeting this goal, the present thesis systematically studies resource allocation tasks for future sustainable cloud networks under uncertainty. With an eye towards realistic scenarios, the thesis progressively adapts elegant mathematical models, optimization frameworks, and develops low-complexity algorithms from three different aspects: stochastic (Chapters 2 and 3), robust (Chapter 4), and big data-driven approaches (Chapter 5). The resultant algorithms are all numerically efficient with optimality guarantees, and most of them are also amenable to a distributed implementation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In the new era of big data analytics, cloud computing, and Internet of Things, data centers are proliferating globally to provide important Internet services such as instant messaging, video distribution, and data backup. For the purposes of reliability and quality-of-service (QoS), a cloud service provider typically owns multiple data centers (DCs) geographically distributed across areas. For instance, Google currently operates seven DCs in the US, and fourteen all over the world [3]. Along with the ever-increasing demand for Internet applications, energy-consuming DCs incur surprisingly high electricity bills. 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 [1]. DCs in the US consumed about 91 billion kWh electricity in 2013, which is almost twice the amount of power needed by all households in New York City [77]. Such a consumption is projected to reach 140 billion kWh by 2020; see Fig. 3.1 for a cloud network with geo-distributed DCs and mapping nodes (MNs). 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 over a large area (e.g., a country).

In order to reduce the electricity cost, considerable efforts from both industry and academia have been made over the last decade [61]. The approach has been to mainly reduce the energy usage in DCs, through e.g., dynamic thermal management, server speed scaling, and by dynamically

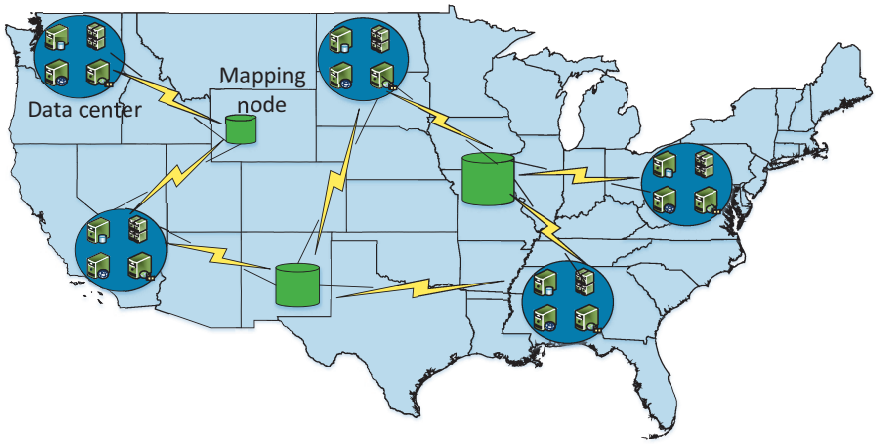

Figure 1.1: A cloud network diagram.

re-sizing the number of active servers [9, 46, 78]. However, not only reducing the electricity cost is of great interest, but also improving the sustainability and efficiency of data centers is essential. Fortunately, contemporary advances of power grid networks can be utilized by the DC infrastructure. Sustainable microgrids, as physical-level supplies of renewable energy sources, energy storage units, and possibly controllable loads, are frequently integrated in the design of current DCs; see also Fig. 1.2 for a smart-grid powered sustainable data center.

Though promising, full benefits of renewable energy generation (e.g., wind and solar energy) can only be harnessed by properly mitigating its intrinsically stochastic nature, which however presents a formidable challenge. Toward addressing this challenge, the virtues of modern big data analytics and machine learning tools are expected to offer major improvements in energy management schemes. For this reason, the present thesis focuses on resource allocation for data center networks under uncertainty. Specifically, the thesis proposes practical and mathematically tractable frameworks yielding low-complexity algorithms from three different aspects: stochastic (Chapters 2 and 3), robust (Chapter 4), and big data-aided optimization (Chapter 5). Most proposed algorithms are also amenable to distributed implementations. The motivation, context, and contributions of this thesis are outlined in the ensuing sections.

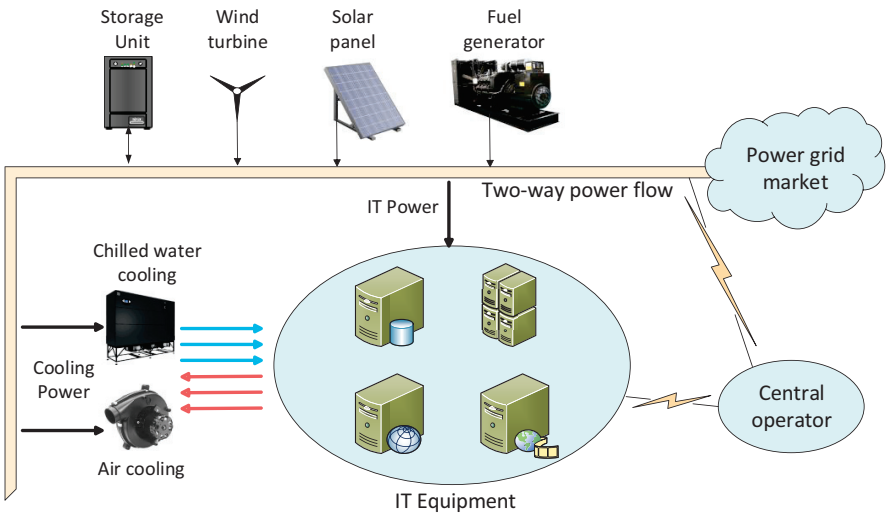

Figure 1.2: A smart-grid powered sustainable data center.

1.2 Cooling-aware Energy Management for Sustainable DCs

Most existing efforts toward sustainable DCs ignore cooling power consumption, despite the fact that a substantial amount of energy in DC goes to their cooling systems [67]. Cooling structures were accounted for in the joint energy and workload management of [48] and [33]. Assuming that the future workload and renewable energy sources (RES) information is known a-priori, [48] investigated energy and workload management *offline*. On the one hand, the computational complexity in [48] can become prohibitively high as the scheduling horizon grows large. On the other hand, future RES and information technology (IT) workloads are generally hard to predict accurately. Online energy and workload management was addressed in [33], using a simplified single source cooling and power supply structure. However, neither [48] nor [33] considered a two-way energy trading mechanism for the DC to potentially sell its surplus energy to the market at a fair price in order to lower operating costs.

Chapter 2 in this thesis considers a practical data center design consisting of power supply, cooling, and IT operating systems. The power supply system comprises a conventional generator, RES, distributed energy storage units, and a mechanism to perform two-way energy trading with the external electricity market. While the cooling system combines two subsystems with different cooling coefficients, the IT operating system can intelligently schedule the workloads under

quality-of-service (QoS) constraints. In this context, an *online* energy and workload management approach is developed in Chapter 2, which dynamically makes *instantaneous* decisions without a-priori knowledge of any statistics of the underlying random workload, renewable, and electricity price processes. To this end, the intended task is formulated as an infinite time horizon optimization problem aiming to minimize the time-average operational network cost. Targeting a low-complexity online solution, we adopt relaxation techniques to decouple the decision variables across time. Then leveraging Lagrange relaxation and stochastic approximation techniques, we develop a novel online control algorithm. Based on the revealed characteristics of the optimal schedules, we formally establish that when the storage device has sufficiently high capacity, or, when the difference between electricity purchase and selling prices is small, the proposed algorithm yields a feasible and near-optimal resource management strategy for the original problem.

1.3 Distributed Stochastic Load Balancing with Incentive Payment

Optimal energy and workload management for setups with a *single* DC have been investigated in Chapter 2 and elsewhere [46,48,72,78]. However, these approaches process user requests locally or presume that optimal routing has been performed, thus missing to account for the spatio-temporal diversity of RES, workload demand and energy prices. Algorithms in [33, 64, 82, 83, 85] either schedule delay-tolerant workloads (DWs) scheduling, or route interactive workloads (IWs) routing task over a DC network. Generalization to a hybrid workload management scheme essentially requires multi-timescale decision making, which is not straightforward.

Chapter 3 of this thesis considers the joint workload and energy management for a cloud network consisting of multiple geo-distributed MNs and DCs. Compared to [33, 64, 82, 83, 85], the proposed workload routing and scheduling policy includes both DWs and IWs, and the energy management scheme integrates the RES, storage units and two-way energy trading, to minimize the total energy cost from cooling and IT operating systems. Furthermore, leveraging the flexibility provided by the new demand-response programs [49, 88], an incentive payment mechanism is developed to modulate the peaks of the IW demand, while guaranteeing the Service Level Agreement (SLA), is implemented; see also [83] and references therein.

Targeting a space-time decoupled online solver, this thesis develops a novel two-timescale al-

gorithm that first dualizes the time-coupling constraints and then, for each time instant, dualizes the constraints coupling MN variables with DC variables. This allows one to solve the problem separately across time and space, provided that the corresponding Lagrange multipliers are updated and exchanged in a coordinated manner. The multipliers of the time-coupling constraints are updated using a stochastic approximation iteration running at the slow timescale, while the multipliers of the MN-DC coupling constraints are updated using the fast iterative shrinkage-thresholding algorithm (FISTA) [12] at a fast timescale. The novel algorithm incurs a low communication overhead, and can be implemented in a distributed fashion. If the random process involved is either independent and identically distributed (i.i.d.) or follows a finite-state ergodic Markov chain, it is further established that the proposed schemes yield a feasible and asymptotically optimal resource-management strategy for the original problem.

1.4 Robust Resource Allocation over Data Center Networks

Existing approaches to dealing with RES uncertainty include the scenario-based stochastic optimization schemes using RES samples from historical data or a given distribution [58, 84], and the stochastic approximation-based schemes; e.g., those in Chapters 2-3 and in [25, 32, 72, 85]. To guarantee convergence and optimality, these methods typically assume i.i.d. RES samples, which can be unrealistic in practice [54, 68].

Chapter 4 of the present thesis considers robust workload and energy management for a cloud network. Distinct from existing works, a *deterministic* uncertainty set of the *unknown* renewable generation, as well as a two-way energy trading mechanism is introduced to account for the stochastic and nondispatchable nature of RES. The proposed uncertainty set of the RES generation only requires easy-to-obtain first-order statistics, and sample correlation statistics. Control parameters are further designed to trade off robustness for conservatism of the robust optimization formulation.

Built on practical models, the resource allocation task is formulated as a robust optimization problem, which minimizes the system's worst-case net cost subject to DCs' operational constraints. Leveraging the problem structure, we show that it can be cast as a convex program. Capitalizing on the dual decomposition approach, an efficient distributed solver is developed. It is shown that the proposed algorithm is guaranteed to yield the desired robust workload and energy-management

strategy, and could also facilitate distributed implementations among the MNs and DCs. Finally, extensive numerical results with real data corroborate the merits of the proposed framework and approaches.

1.5 Learning-aided Resource Allocation over Data Center Networks

Turning attention to algorithmic issues of our stochastic resource allocation task, the main limitation of existing stochastic schemes is their slow convergence speed and the high network delay (high battery capacity requirement) as a by-product. Facing this limitation, several recent attempts have been taken to come up with first- and second-order optimization algorithms [44,47,86,87]. However, design philosophy of Chapter 5 is distinct from that in [44, 47, 86, 87] in the sense that the goal is to develop a comprehensive learning protocol to integrate statistical learning approches into stochastic resource allocation tasks, solved using nonlinear optimization tools.

Targeting this goal, an interesting observation is that renowned algorithms from the body of network optimization literature (e.g., backpressure and maxweight) are implicitly connected with the Lagrange dual theory, and the important role of Lagrange multipliers has been frequently highlighted [37, 38, 73]. Looking forward, the proliferation of machine learning advances motivates a systematic way to uncover "hidden insights" through learning from historical relationships and trends in massive dataset [18, 75]. In this context, Chapter 5 of the present thesis revisits the stochastic network optimization problem from a machine learning vantage point with a goal of learning the system Lagrange multiplier in a fast and efficient manner. Unfortunately, designing online resource allocation algorithms capitalizing on data-driven learning schemes is still an open problem. The most relevant work in this direction is [37]. However, building upon the Lyapunov optimization framework, the per-iteration learning scheme in [37] involves constructing a histogram to estimate the underlying distribution. While acceptable for a discrete probability distribution with finite support, an non-negligible quantization error or a prohibitively high computational complexity is unavoidable for a continuous distribution.

In Chapter 5, the network resource allocation problem is explicitly formulated as a learning task that entails batch training and online testing. In the batch training mode, by recognizing that the problem is in the form of maximizing the finite sum of concave functions, we connect it with a

prevalent machine learning routine called empirical risk minimization (ERM) [75]. Leveraging this batch problem structure, we modify the recently developed stochastic average gradient approach (SAGA) to fit our training setup, and efficiently compute an empirical Lagrange multiplier with an order-optimal convergence rate at a fairly low computational cost per iteration. In the online testing mode, a novel dynamic resource allocation approach (that we term online SAGA) is proposed. It operates in a *LeArning-while-TestINg* (Latin) fashion. The online SAGA can be viewed as a combination of *stochastic approximation* and *statistical learning*: in the learning phase, it preserves the simple update of offline SAGA to dynamically learn from streaming data and maintain the learning error always below the statistical accuracy; while in the testing phase, it incorporates merits of the well-appreciated stochastic dual subgradient (SDGD) to explicitly track the queue variations, and guarantee the long-term queue stability.

Chapter 2

Cooling-aware Energy Management for Sustainable Data Centers

2.1 System Models

Consider a data center composed of three subsystems: the IT system, the cooling system coping with the heat generated by the IT system, and the power supply system supporting IT and cooling equipments.

2.1.1 Workload model

In general, workloads in data centers fall under two categories: delay-sensitive (or 'must-serve') and delay-tolerant workloads [85]. The first category includes voice and multimedia services, as well as real-time user requests, which have to be served usually within a few seconds. Delay-tolerant workloads include HTTP and email deliveries that can be scheduled to run when the energy cost is low, or, when the system workload is low. This second category provides ample optimization opportunities for workload management adaptive to the time-varying amounts of RES and cooling supply.

Consider an infinite scheduling horizon, indexed by the set $\mathcal{T} := \{0, 1, 2, \ldots\}$, and suppose that there are I types of 'must-serve' workloads with the central operator having to allocate IT capacity $\check{v}_i^t$ per slot t for type i. On the other hand, suppose that there are J classes of delay-tolerant

workloads, where workloads in class j have total demand $\check{w}_j^t$ at slot t and maximum parallelization MP_j. With w_j^t denoting the IT capacity allocated to the delay-tolerant workloads in class j at slot t, it must hold that

$$0 \leq w_j^t \leq \min\{\check{w}_j^t, MP_j\}, \ \ \forall t \tag{2.1}$$

and the total IT demand (consumption) at slot t is given by

$$d^t = \sum_{i=1}^{I} \check{v}_i^t + \sum_{j=1}^{J} w_j^t, \ \ \forall t. \tag{2.2}$$

Supposing that the total IT capacity is $D_{\text{IT}} > \sum_{i=1}^{I} \check{v}_i^t$, the per-slot IT demand should clearly satisfy

$$0 \leq d^t \leq D_{\text{IT}}, \quad \forall t. \tag{2.3}$$

In order to accommodate QoS requirements, a limiting time-average constraint is also introduced to bound the fraction of pending delay-tolerant requests; that is,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\check{w}_j^t - w_j^t}{\check{w}_j^t} \leq \eta, \quad \forall j \tag{2.4}$$

where η is a prescribed threshold. We will assume that unserved requests or their fractions will be automatically requested in the ensuing slot(s).

With $\check{\mathbf{v}}^t := [\check{v}_1^t, \ldots, \check{v}_I^t]^\top$, and likewise for $\check{\mathbf{w}}^t$ and $\mathbf{w}^t$, assume for simplicity that random processes $(\check{\mathbf{v}}^t, \check{\mathbf{w}}^t)$ are independently and identically distributed (i.i.d.) across time. Under (2.1)-(2.4), the IT system variables to optimize are $\{\mathbf{w}^t\}$.

2.1.2 Cooling structure

Along with the increasing density of IT equipment in data centers, a considerable amount of electricity is consumed by the cooling system that generally operates in two modes [9, 48]: outside-air (OA) and chilled-water (CW) cooling.

The energy usage of *outside-air cooling* is mainly the power consumed by blowers, which can be approximated as a cubic function of the blower speed [91]. From basics of heat transfer and the general fan laws, it turns out that the blower speed under tight control is proportional to the IT

demand d^t [36]. As a result, the outside-air cooling power consumption can be modeled as a convex function of d^t, namely

$$f_{\rm OA}(d^t) = \kappa_1 (d^t)^3, \;\; 0 \leq d^t \leq \check{d}_{\rm OA} \tag{2.5}$$

where $\kappa_1 > 0$ depends on the temperature difference between the (hot) exhausting air temperature $T_{\rm RA}$ from the IT racks and the outside air temperature $T_{\rm OA}$. The maximum capacity of outside-air cooling in (4.8) can be modeled as $\check{d}_{\rm OA} = C(T_{\rm RA} - T_{\rm OA})$, with $C > 0$ proportional to the maximal outside air mass flow rate. Clearly, the cooling efficiency of outside-air cooling is greatly affected by the air temperature. As a consequence, this approach is usually complemented by more stable cooling resources, such as chillers.

The *chilled-water cooling* model here is built on the actual measurement of an operational chiller whose power consumption can be approximated as [59]

$$f_{\rm CW}(d^t) = \kappa_2 d^t \tag{2.6}$$

where d^t is again the IT demand in (4.6), and $\kappa_2 > 0$ is a constant depending on the specific chiller characteristics.

Clearly, the two approaches have different cooling efficiencies and capacities, which provides the possibility to optimize the power consumption for cooling by properly combing these decoupled sources. In particular, for a given d^t, there is an optimal allocation between air- and water-based cooling. Let $d^t_{\rm CW}$ and $d^t_{\rm OA}$ denote the amounts of IT demand allocated for water and air cooling, respectively. The optimal cooling power consumption is (cf. (4.8) and (4.9) with $d^t_{\rm CW} = d^t - d^t_{\rm OA}$)

$$f(d^t) = \min_{0 \leq d^t_{\rm OA} \leq \check{d}_{\rm OA}} \kappa_2 [d^t - d^t_{\rm OA}]^+ + \kappa_1 (d^t_{\rm OA})^3. \tag{2.7}$$

Letting $d_s := \min\{\check{d}_{\rm OA}, \sqrt{\kappa_2/(3\kappa_1)}\}$, the convex problem in (4.10) can be solved in closed form

$$f(d^t) = \begin{cases} \kappa_1 (d^t)^3, & d^t \leq d_s \\ \kappa_1 d_s^3 + \kappa_2 (d^t - d_s), \; \text{otherwise} \end{cases} \tag{2.8}$$

with the optimal demands split between cooling models as

$$(d^t_{\rm OA})^* = \begin{cases} d^t, & d^t \leq d_s \\ d_s, & \text{otherwise} \end{cases} \tag{2.9}$$

and

$$(d_{\mathrm{CW}}^t)^* = d^t - (d_{\mathrm{OA}}^t)^*. \tag{2.10}$$

Note that κ_1 and $\check{d}_{\mathrm{OA}}$, and thus d_s, as well as $f(d^t)$ in (4.1.2) are random. And it is worth stressing that although we adopt a specific cooling model here, our approach applies to any nondecreasing and convex function $f(d^t)$ in (4.1.2).

2.1.3 Power supply model

Consider a data center supplied by a RES-integrated microgrid consisting of a conventional generator (CG) (e.g., fuel generator), an on-site renewable generator (RG) (e.g., wind or solar), and N distributed energy storage units (e.g., batteries) [63, 70]. The distributed storage units in this model can include batteries deployed at renewable generators, batteries in electric vehicles, and uninterrupted power supply (UPS) units inside the data center itself; see e.g., [45]. Since the considered energy management task is within a geographically small area (e.g., a microgrid around a data center), the cost of moving energy is deemed negligible.

Let P_c^t denote the energy output of the CG per slot t upper bounded by $P_c^{\max}$; that is,

$$0 \leq P_c^t \leq P_c^{\max}, \ \ \forall t. \tag{2.11}$$

The change of the CG energy outputs in two consecutive slots is bounded by the following so-termed ramping constraints:

$$P_c^t - P_c^{t-1} \leq R_{\mathrm{up}}, \quad P_c^{t-1} - P_c^t \leq R_{\mathrm{dw}} \tag{2.12a}$$

where R_{up} and R_{dw} are known maximum ramping-up and ramping-down rates. In particular, if $R_{\mathrm{up}} = R_{\mathrm{dw}} = \rho P_c^{\max}$, the ramping constraints can be compactly expressed as

$$|P_c^t - P_c^{t-1}| \leq \rho P_c^{\max}, \ \ \forall t \tag{2.12b}$$

where $\rho \in [0, 1]$ reflects tightness of the ramping requirements.

The renewable energy r^t generated from the on-site RG per slot t is assumed i.i.d. across slots to simplify performance analysis. But as will be seen in our simulated tests, the proposed algorithm remains operational without any modification to non-i.i.d. $\{r^t\}$ processes too. Yet, performance

guarantees in the non-i.i.d. case require more elaborate multi-slot Lyapunov drift techniques along the lines of [54].

Let C_n^0 and C_n^t denote the initial amount of stored energy and the state of charge (SoC) in the n-th storage unit at the beginning of time slot t. Each unit has finite capacity $C_n^{\max}$. Furthermore, for reliability purposes, it may be required to ensure that a minimum energy level $C_n^{\min}$ is maintained at all times[1]; this necessitates the two-sided inequalities

$$C_n^{\min} \leq C_n^t \leq C_n^{\max}, \ \ \forall n, t. \tag{2.13}$$

Let $P_{b,n}^t$ denote the power delivered to or drawn from the n-th storage unit (battery) at slot t, which amounts to either charging ($P_{b,n}^t > 0$) or discharging ($P_{b,n}^t < 0$). Hence, the stored energy obeys the dynamic equation

$$C_n^{t+1} = C_n^t + P_{b,n}^t, \ \ \forall n, t. \tag{2.14}$$

The amount of power (dis)charged is bounded by

$$P_{b,n}^{\min} \leq P_{b,n}^t \leq P_{b,n}^{\max}, \ \ \forall n, t \tag{2.15}$$

where $P_{b,n}^{\min} < 0$ and $P_{b,n}^{\max} > 0$ are set by physical limits.

Overall, the total consumption P_{out}^t of the data center per slot t includes the IT demand d^t, the cooling power consumption $f(d^t)$, and the charged power $P_{b,n}^t > 0$; that is,

$$P_{\text{out}}^t = d^t + f(d^t) + \sum_{n=1}^{N} P_{b,n}^t \mathbb{1}_{(P_{b,n}^t > 0)}. \tag{2.16}$$

Likewise, the total energy supply P_{in}^t per slot t is given by

$$P_{\text{in}}^t = P_c^t + r^t - \sum_{n=1}^{N} P_{b,n}^t \mathbb{1}_{(P_{b,n}^t < 0)}. \tag{2.17}$$

Besides the IT variables $\{\mathbf{w}^t\}$, under constraints (4.11)-(4.19), the power supply variables to optimize are CG and battery power amounts $\{P_c^t, \mathbf{p}_b^t\}$, where $\mathbf{p}_b^t := [P_{b,1}^t, \ldots, P_{b,N}^t]^\top$.

---

[1]Storage devices become unreliable with high depth-of-discharge (DoD) – percentage of maximum charge removed during a discharge cycle; hence, a minimum level $C_n^{\min}$ can avoid high DoD. Such a level can also support the data center operation in the event of a grid outage.

2.1.4 Cost-revenue model

In addition to the internal energy resources (namely, CG, RG, storage units), the data center can resort to the external energy markets in an on-demand manner. With a two-way energy trading facility, the data center can buy energy from the external energy markets when in a deficit ($P^t_{\rm out} > P^t_{\rm in}$), or, sell energy to the markets in the case of a surplus ($P^t_{\rm out} < P^t_{\rm in}$). Clearly, the shortage energy purchased by the data center is $[P^t_{\rm out} - P^t_{\rm in}]^+$; while the surplus energy that can be sold is $[P^t_{\rm in} - P^t_{\rm out}]^+$. Both the shortage and surplus energies are non-negative, and at most one of them is positive per time slot t.

Let α^t_c denote per unit the CG cost at slot t. Suppose that the energy can be bought from the external energy markets at price α^t_b, while the energy is sold to them at price α^t_s per slot t. Notwithstanding, we shall always set $\alpha^t_b \geq \alpha^t_s$ to avoid less relevant buy-and-sell activities of the data center for profit.

Again, we will suppose for simplicity that the prices $(\alpha^t_c, \alpha^t_b, \alpha^t_s)$ are random i.i.d. over time. Per slot t, the energy transaction *cost* for the data center is therefore

$$\mathcal{C}(\mathbf{w}^t, \mathbf{p}^t_b, P^t_c) := \alpha^t_c P^t_c + \alpha^t_b [P^t_{\rm out} - P^t_{\rm in}]^+ - \alpha^t_s [P^t_{\rm in} - P^t_{\rm out}]^+. \tag{2.18}$$

Note that a linear cost of CG is introduced only to simplify the proofs in Section IV. Any convex and Lipschitz continuous cost could replace the linear one and lead to similar results.

Since the *revenue* from 'must-serve' workloads $\check{\mathbf{v}}^t$ is fixed, we account only for the revenue from the delay-tolerant workloads. Specifically, the revenue per slot t is given by

$$\mathcal{R}(\mathbf{w}^t) := \textstyle\sum_{j=1}^J u_j w^t_j \tag{2.19}$$

where u_j is the revenue per unit of workloads in class j, and $u_j w^t_j$ captures the total revenue of class-j delay-tolerant workloads earned per slot t. (Here too, any concave function could replace the linear combination in (2.19).)

At this point, it is instructive to collect all sources of randomness into the state vector defined as

$$\boldsymbol{\sigma}_t := [(\check{\mathbf{v}}^t)^\top, (\check{\mathbf{w}}^t)^\top, \kappa_1, \check{d}_{\rm OA}, \alpha^t_c, \alpha^t_b, \alpha^t_s, r^t]^\top \tag{2.20}$$

and also the optimization variable into the vector

$$\mathbf{x}_t := [(\mathbf{w}^t)^\top, (\mathbf{p}^t_b)^\top, P^t_c]^\top = \chi(\boldsymbol{\sigma}_t) \tag{2.21}$$

where the last equality denotes the control strategy $\chi(\cdot)$ that depends on the state $\boldsymbol{\sigma}_t$ to output the settings $\mathbf{x}_t$ per slot.

2.2 Dynamic Energy and Workload Management

Based on the models of Section II, we pursue in this section optimal power and workload management of a data center, starting with the operational *net-cost* per slot t that is given by [cf. (2.18) and (2.19)]

$$\Psi(\mathbf{x}_t) := \mathcal{C}(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t) - \mathcal{R}(\mathbf{w}^t). \tag{2.22}$$

Random process $\{\Psi(\mathbf{x}_t)\}$ is generally nonstationary. Besides $\boldsymbol{\sigma}_t$, the nonstationarity of $\{\Psi(\mathbf{x}_t)\}$ is also due to the time-varying $\{C_n^t\}$, which affects (dis)charging decisions $P_{b,n}^t$. However, as numerical tests will also corroborate, $\{\Psi(\mathbf{x}_t)\}$ can be safely assumed mean ergodic in several practical settings; that is, limiting time averages involving $\{\Psi(\mathbf{x}_t)\}$ will be henceforth assumed to exist in the appropriate sense[2].

Over the scheduling horizon, the central operator of the data center seeks an optimal schedule for flexible workloads $\{\mathbf{w}^t\}$, CG energy generation $\{P_c^t\}$, and battery charging energy $\{\mathbf{p}_b^t\}$, in order to minimize the limiting time-averaged net-cost, subject to IT operation constraints as well as

[2]Depending on so-termed mixing conditions assumed, the convergence of limits can be in probability, mean-square sense, or, almost surely (as).

energy generation and storage constraints. Concretely written, we wish to solve

$$\Psi^* := \min_{\{\mathbf{x}_t\}_t} \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \Psi(\mathbf{x}_t) \tag{2.23a}$$

$$\text{s. t.} \quad C_n^{t+1} = C_n^t + P_{b,n}^t,\ \forall n,t \tag{2.23b}$$

$$C_n^{\min} \le C_n^t \le C_n^{\max},\ \forall n,t \tag{2.23c}$$

$$P_{b,n}^{\min} \le P_{b,n}^t \le P_{b,n}^{\max},\ \forall n,t \tag{2.23d}$$

$$0 \le P_c^t \le P_c^{\max},\ \forall t \tag{2.23e}$$

$$|P_c^t - P_c^{t-1}| \le \rho P_c^{\max},\ \forall t \tag{2.23f}$$

$$0 \le w_j^t \le \min\{\check{w}_j^t, MP_j\},\ \forall j,t \tag{2.23g}$$

$$d^t = \textstyle\sum_{i=1}^I \check{v}_i^t + \sum_{j=1}^J w_j^t,\ \forall t \tag{2.23h}$$

$$0 \le d^t \le D_{\mathrm{IT}},\ \forall t \tag{2.23i}$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \frac{\check{w}_j^t - w_j^t}{\check{w}_j^t} \le \eta,\ \forall j \tag{2.23j}$$

where the instantaneous constraints (4.21d)-(4.21c) involving random variables are understood to hold almost surely.

For the net-cost $\Psi(\mathbf{x}_t)$, we can establish the following.

Lemma 1. *Viewed as a deterministic function,* $\Psi(\mathbf{x}_t)$ *per realization is jointly convex in* $(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t)$.

Proof. With $\delta_-^t := (\alpha_b^t - \alpha_s^t)/2$ and $\delta_+^t := (\alpha_b^t + \alpha_s^t)/2$, it follows readily from (2.18) that

$$\mathcal{C}(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t) = \delta_-^t |P_{\mathrm{out}}^t - P_{\mathrm{in}}^t| + \delta_+^t (P_{\mathrm{out}}^t - P_{\mathrm{in}}^t) + \alpha_c^t P_c^t.$$

Since $\delta_-^t \ge 0$, it is clear that $\mathcal{C}(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t)$ is a convex and nondecreasing function of $P_{\mathrm{out}}^t - P_{\mathrm{in}}^t$. Recall that $P_{\mathrm{out}}^t - P_{\mathrm{in}}^t = d^t + f(d^t) + \sum_{n=1}^N P_{b,n}^t - P_c^t - r^t$ and $d^t = \sum_{i=1}^I \check{v}_i^t + \sum_{j=1}^J w_j^t$. Given that $f(d^t)$ is convex, it is easy to see that $\mathcal{C}(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t)$ is jointly convex in $(d^t, \mathbf{p}_b^t, P_c^t)$ [19, Chapter 3.2]. As d^t is an affine transformation of $\mathbf{w}^t$, it follows that $\mathcal{C}(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t)$ is jointly convex in $(\mathbf{w}^t, \mathbf{p}_b^t, P_c^t)$; and so is $\Psi(\mathbf{x}_t)$. □

2.2.1 Problem relaxation

As the cost in (5.7a) is convex per Lemma 1 and all the constraints are linear, problem (5.7) is a convex program. However, it is still impossible to solve due to the infinite time horizon. Furthermore, the battery SoC dynamic equations (4.21b) and the CG ramping constraints (2.23f) couple the optimization variables over the infinite time horizon. This renders traditional solvers, such as dynamic programming, intractable.

To turn (5.7) into a tractable form, we adopt queue-based relaxation techniques [42, 50, 72], by recognizing that SoC dynamics in (4.21b) can be viewed as charge-based queue recursions; see also [29]. For the random state $\boldsymbol{\sigma}_t$, we assume that mean ergodicity holds in the appropriate sense e.g., almost surely (as), meaning

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\Psi(\mathbf{x}_t)\overset{\text{as}}{=}\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Psi(\chi\left(\boldsymbol{\sigma}_t\right))]:=\overline{\Psi} \tag{2.24}$$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}P_{b,n}^t\overset{\text{as}}{=}\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[P_{b,n}^t\right]:=\overline{P}_{b,n},\forall n \tag{2.25}$$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\frac{\check{w}_j^t-w_j^t}{\check{w}_j^t}\overset{\text{as}}{=}\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\frac{\check{w}_j^t-w_j^t}{\check{w}_j^t}\right]:=\Delta\overline{w}_j,\ \forall j \tag{2.26}$$

where expectations are over the distribution of $\boldsymbol{\sigma}_t$, and the possible randomness of the control policy.

Instead of the original problem (5.7), we thus aim at the functional optimization problem

$$\overline{\Psi}^*:=\min_{\chi(\cdot)}\ \overline{\Psi} \tag{2.27a}$$

$$\text{s. t.}\quad \overline{P}_{b,n}=0,\ \forall n \tag{2.27b}$$

$$\Delta\overline{w}_j-\eta\leq 0,\ \forall j \tag{2.27c}$$

$$(3.15d)-(3.15e),(4.21d)-(4.21e)$$

where $\chi(\cdot)$ denotes the mapping (function) from the random state $\boldsymbol{\sigma}_t$ to the vector $\mathbf{x}_t$ of optimization variables.

Comparing (2.27) with (5.7), constraints (4.21b)-(3.15c) have been replaced by the time-average constraints (2.27b), and variables $\{C_n^t\}$ have been eliminated. In addition, the time-coupled ramping constraints (2.23f) are removed and the QoS constraints (4.21c) are re-written compactly. We contend that (2.27) is a relaxed version of (5.7). To recognize this, take any schedule $\{\mathbf{w}^t,\mathbf{p}_b^t,P_c^t\}_{t=0}^{T-1}$

that satisfies (4.21b)-(3.15c) in (5.7). Then summing (4.21b) over time and taking expectation yields $\mathbb{E}[C_n^{T-1}] = \mathbb{E}[C_n^0] + \sum_{t=0}^{T-1} \mathbb{E}[P_{b,n}^t]$, $\forall n$. Since both C_n^0 and C_n^{T-1} are bounded due to (3.15c), dividing both sides by T and taking limits as $T \to \infty$, implies (2.27b). As constraints (2.23f) are simply ignored in (2.27), it is clear that any feasible schedule for (5.7) is also feasible for (2.27). This implies that (2.27) is a relaxation of (5.7), which in turn establishes that $\overline{\Psi}^* \leq \Psi^*$.

With the time-coupled constraints relaxed, (2.27) appears more tractable than (5.7). Specifically, it can be shown that the optimal solution to (2.27) is achieved by a time-invariant (generally stationary) control policy $\chi(\cdot)$ that chooses per-slot variables $\mathbf{x}_t$ purely as a function (possibly randomized) of the current state $\boldsymbol{\sigma}_t$, regardless of the storage energy C_n^t [54, Theorem 4.5]. As a consequence, a stochastic dual subgradient solver is developed for (2.27) next, which under proper initialization yields a feasible and near-optimal solution of (5.7).

2.2.2 Lagrange dual approach

Consider the feasible set $\mathcal{X}_t$ arising due to the instantaneous constraints of (2.27) as

$$\mathcal{X}_t := \{\mathbf{x}_t \mid \mathbf{x}_t \text{ satisfying } (3.15d) - (3.15e), \text{ and } (4.21d) - (4.21e)\}.$$

Let $\boldsymbol{\lambda} := [\lambda_1, \ldots, \lambda_N]^\top$ and $\boldsymbol{\nu} := [\nu_1, \ldots, \nu_J]^\top$ denote the Lagrange multipliers associated with the constraints (2.27b) and (2.27c), respectively. With the compact notation $\mathbf{x} := \{\mathbf{x}_t, \forall t\}$, and $\boldsymbol{\pi} := [\boldsymbol{\lambda}^\top, \boldsymbol{\nu}^\top]^\top$, the partial Lagrangian of (2.27) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\pi}) := \overline{\Psi} + \sum_{n=1}^{N} \lambda_n \overline{P}_{b,n} + \sum_{j=1}^{J} \nu_j \left(\Delta \overline{w}_j - \eta\right) \tag{2.28}$$

while the Lagrange dual function is given by

$$\mathcal{D}(\boldsymbol{\pi}) := \min_{\{\mathbf{x}_t \in \mathcal{X}_t\}_t} \mathcal{L}(\mathbf{x}, \boldsymbol{\pi}) \tag{2.29}$$

and the dual problem of (2.27) is: $\max_{\boldsymbol{\nu} \geq \mathbf{0}, \boldsymbol{\lambda}} \mathcal{D}(\boldsymbol{\pi})$.

For the dual problem, a standard subgradient iteration can be employed to obtain the optimal $\boldsymbol{\pi}^*$, namely

$$\lambda_n(k+1) = \lambda_n(k) + \mu g_{\lambda_n}(k) \quad , \forall n \tag{2.30a}$$

$$\nu_j(k+1) = \left[\nu_j(k) + \mu g_{\nu_j}(k)\right]^+ , \forall j \tag{2.30b}$$

where k is the iteration index; $\mu > 0$ is a constant stepsize; and $g_{\lambda_n}(k)$ and $g_{\nu_j}(k)$ denote the subgradients of (5.12) with respect to λ_n and ν_j, expressed as

$$g_{\lambda_n}(k) = \overline{P}_{b,n}(\boldsymbol{\pi}(k)) \tag{2.31a}$$

$$g_{\nu_j}(k) = \Delta\overline{w}_j(\boldsymbol{\pi}(k)) - \eta \tag{2.31b}$$

where $\overline{P}_{b,n}(\boldsymbol{\pi}(k))$ and $\Delta\overline{w}_j(\boldsymbol{\pi}(k))$ denote the primal variables given by the minimization of (5.11) over $\mathbf{x}$ for $\boldsymbol{\pi} = \boldsymbol{\pi}(k)$. Due to the linearity of the limiting average and the expectation in $\overline{\Psi}$, $\overline{P}_{b,n}$, and $\Delta\overline{w}_j$, these operations can be interchanged with the minimization of $\mathcal{L}$ in (5.11). Accordingly, $\overline{P}_{b,n}(\boldsymbol{\pi}(k))$ and $\Delta\overline{w}_j(\boldsymbol{\pi}(k))$ can be found by solving the following minimization over the infinite horizon [cf. (2.22)]

$$\begin{aligned}\left\{P^t_{b,n}(\boldsymbol{\pi}(k)), w^t_j(\boldsymbol{\pi}(k))\right\} \in \arg \min_{\mathbf{x}_t \in \mathcal{X}_t} \Psi(\mathbf{x}_t) \\ + \sum_{n=1}^{N} \lambda_n(k) P^t_{b,n} + \sum_{j=1}^{J} \nu_j(k) \left(1 - \eta - \frac{w^t_j}{\check{w}^t_j}\right).\end{aligned} \tag{2.32}$$

Note that $P^t_c(\boldsymbol{\pi}(k))$ will be obtained from (2.32) as well, but it may be infeasible for the original problem (5.7) since the ramping constraint (2.23f) is not included in the feasible set $\mathcal{X}_t$.

Since $\mathcal{X}_t$ is a convex set and the objective is a convex function of $\{P^t_{b,n}, w^t_j\}$, the minimization in (2.32) is a convex program that can be efficiently solved to obtain the minimizer $\{P^t_{b,n}(\boldsymbol{\pi}(k)), w^t_j(\boldsymbol{\pi}(k))\}$. The multiplier iterations (2.30) are guaranteed to converge to a neighborhood of the optimal multipliers $\boldsymbol{\pi}^*$ for the dual problem [15, Section 6.3].

A challenge associated with (4.27) is computing $\overline{P}_{b,n}(\boldsymbol{\pi}(k))$ and $\Delta\overline{w}_j(\boldsymbol{\pi}(k))$ per iteration k. This requires performing (high-dimensional) integration over the unknown multivariate distribution function of $\boldsymbol{\sigma}_t$; and approximately, finding the corresponding limiting time-averages in (2.24)-(2.26), both of which are impractical. To circumvent this impasse, a stochastic subgradient approach is devised next to find the stochastic estimates 'on-the-fly' [50, 60].

2.2.3 Stochastic approximation solver

Consider dropping the expectations in (4.27) and merging indices k and t, to arrive at the corresponding stochastic iterations [cf. (2.30)]

$$\hat{\lambda}_n^{t+1} = \hat{\lambda}_n^t + \mu P_{b,n}^t(\hat{\boldsymbol{\pi}}_t),\ \forall n \tag{2.33a}$$

$$\hat{\nu}_j^{t+1} = \left[\hat{\nu}_j^t + \mu\left(1 - \eta - \frac{w_j^t(\hat{\boldsymbol{\pi}}_t)}{\breve{w}_j^t}\right)\right]^+,\ \forall j \tag{2.33b}$$

where $\hat{\boldsymbol{\lambda}}_t := [\hat{\lambda}_1^t, \ldots, \hat{\lambda}_N^t]^\top$, and $\hat{\boldsymbol{\nu}}_t := [\hat{\nu}_1^t, \ldots, \hat{\nu}_J^t]^\top$ denote the stochastic estimates of the Lagrange multipliers in (2.30); and $\hat{\boldsymbol{\pi}}_t := [\hat{\boldsymbol{\lambda}}_t^\top, \hat{\boldsymbol{\nu}}_t^\top]^\top$. Given $\hat{\boldsymbol{\pi}}_t$, variables $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t)$ and $w_j^t(\hat{\boldsymbol{\pi}}_t)$ are obtained by solving for [cf. (2.32)]

$$\hat{\Omega}^* := \min_{\{\mathbf{w}^t, \mathbf{p}_b^t, P_c^t\}} \Psi(\mathbf{x}_t) + \sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t + \sum_{j=1}^J \hat{\nu}_j^t \left(1 - \eta - \frac{w_j^t}{\breve{w}_j^t}\right)$$

$$\text{s. t.} \quad (3.15d) - (4.21e). \tag{2.34}$$

In words, (2.33) constitutes an *online* approximation of the batch iterations (2.30) based on the *instantaneous* decisions $\{P_{b,n}^t(\hat{\boldsymbol{\pi}}_t), w_j^t(\hat{\boldsymbol{\pi}}_t)\}$ per slot t. This stochastic approach is made possible thanks to the decoupling of optimization variables across time in (2.27).

Different from (2.32), here the ramping constraints (2.23f) are added back in (2.34). Yet, P_c^{t-1} is not an optimization variable here, but it is treated as a constant determined from the previous slot $t-1$. Clearly, (2.34) is a convex problem per slot t, which can be efficiently solved in polynomial time by existing solvers [5]. The proposed (modified) stochastic subgradient solver is summarized in Algorithm 1. With the addition of (2.23f) in (2.34), the online energy and workload schedule provided by Algorithm 1 is guaranteed to satisfy the physical ramping constraints. Interestingly, it can be shown that the proposed algorithm with proper initialization also yields a schedule that satisfies the storage constraints (4.21b)-(3.15c), and offers a near-optimal solution of the original problem (5.7).

It is worth mentioning that the proposed stochastic solver incurs affordable low computational complexity. Per slot t, the worst-case complexity of solving (2.34) is $\mathcal{O}\left((N+J)^{3.5}\right)$ by interior-point methods [5], while updating (2.33) requires just linear complexity $\mathcal{O}(N+J)$.

Algorithm 1 Online Power and Workload Management

Initialize: with a proper $\hat{\boldsymbol{\pi}}_0$ and stepsize μ

for $t = 1, 2 \ldots$ **do**

Acquire $\boldsymbol{\sigma}_t$, and find $\hat{\boldsymbol{\pi}}_t$ as in (2.33)

Solve (2.34) to obtain instantaneous schedule $\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)$

Perform online operations based on $\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)$ in (2.34)

Update Lagrange multipliers via (2.33)

end for

2.3 Performance Guarantees

To arrive at our main analytical claim, we first establish the optimality gap of the proposed Algorithm 1.

2.3.1 Optimality gap

To begin with, introduce the definition

$$\check{\Omega}^* := \min_{\{\mathbf{w}^t, \mathbf{p}_b^t, P_c^t\}} \Psi(\mathbf{x}_t) + \sum_{n=1}^{N} \hat{\lambda}_n^t P_{b,n}^t + \sum_{j=1}^{J} \hat{\nu}_j^t \left(1 - \eta - \frac{w_j^t}{\check{w}_j^t}\right)$$
$$\text{s. t.} \quad (3.15d) - (3.15e), (4.21d) - (4.21e) \tag{2.35}$$

where $\hat{\boldsymbol{\pi}}_t$ as in (2.34). Compared with (2.34), (2.23f) is absent from (2.35); hence, it clearly holds that $\hat{\Omega}^* \geq \check{\Omega}^*$.

Upon defining $\overline{\alpha}_b := \max\{\alpha_b^t, \forall t\}$ and $\overline{\alpha}_c := \max\{\alpha_c^t, \forall t\}$ [cf. (2.18)], the following lemma can be established.

Lemma 2. *The optimal value of problem (2.34) satisfies*

$$\hat{\Omega}^* \leq \check{\Omega}^* + \delta(\rho)$$

where $\delta(\rho) := (1 - \rho) P_c^{\max} \max\{\overline{\alpha}_b, \overline{\alpha}_c\}$.

Proof. See Appendix A. □

Lemma 2 shows that inclusion of the ramping constraints to subproblem (2.34) will only incur a bounded optimality loss of the stochastic subgradient solver. The proof follows the steps in [70, Theorem 1.2]. Based on this, we can subsequently build on the stochastic optimization techniques in [42, 50, 72] to establish the following lemma.

Lemma 3. *If state $\boldsymbol{\sigma}_t$ is i.i.d. over slots, then the limiting time-average net-cost incurred by the proposed online algorithm satisfies*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right]\leq\Psi^*+\delta(\rho)+\mu M$$

where the constant $M:=\frac{1}{2}J(1+\eta^2)+\frac{1}{2}\sum_{n=1}^{N}(\max\{P_{b,n}^{\max},-P_{b,n}^{\min}\})^2$, and Ψ^* is the optimal value of (5.7) under any feasible control.

Proof. See Appendix B. □

Lemma 7 asserts that the proposed Algorithm 1 converges asymptotically to a region with optimality gap smaller than $\delta(\rho)+\mu M$. The gap approaches a constant $\delta(\rho)$ as the stepsize $\mu\to 0$. In addition, $\delta(\rho)$ can become negligible when the ramping constraints are loose, meaning as ρ approaches 1.

2.3.2 Feasibility guarantee

Lemma 3 established that the proposed scheme can achieve a near-optimal objective value for (5.7). However, since Algorithm 1 is modified from a stochastic solver of the relaxed (2.27), it does not guarantee that the resultant dynamic control policy is a feasible one for (5.7). In the sequel, we will establish that Algorithm 1 indeed yields a feasible policy for (5.7) when it is properly initialized. To this end, we first need the following lemma.

Lemma 4. *With $\underline{\alpha}_s:=\min\{\alpha_s^t,\forall t\}$, the real-time battery (dis)charging decisions $P_{b,n}^t$ returned by the proposed online algorithm obey: i) $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t)=P_{b,n}^{\min}$, if $\hat{\lambda}_n^t>-\underline{\alpha}_s$; or, ii) $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t)=P_{b,n}^{\max}$, if $\hat{\lambda}_n^t<-\overline{\alpha}_b$.*

Proof. See Appendix C. □

Lemma 8 reveals a salient structure of the optimal solution for problem (2.34). Such a structure can be justified by the economic interpretation of the Lagrange multipliers. Specifically, $\hat{\lambda}_n^t$ can be viewed as the stochastic instantaneous charging price. For high prices $\hat{\lambda}_n^t > -\underline{\alpha}_s$, the optimal decision is to discharge the battery as much as possible, i.e., $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = P_{b,n}^{\min}$. Conversely, the battery units can afford full charge $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = P_{b,n}^{\max}$, if the price is low; i.e., $\hat{\lambda}_n^t < -\overline{\alpha}_b$.

Relying on the solution structure revealed by Lemma 8, we can subsequently establish the following lemma.

Lemma 5. *If the stepsize satisfies $\mu \geq \underline{\mu}$, where*

$$\underline{\mu} := \frac{\overline{\alpha}_b - \underline{\alpha}_s}{\min\left\{C_n^{\max} - C_n^{\min} + P_{b,n}^{\min} - P_{b,n}^{\max}, \forall n\right\}} > 0$$

then the proposed algorithm guarantees that the Lagrange multipliers satisfy $\hat{\lambda}_n^t \in \Big[-\overline{\alpha}_b + \mu P_{b,n}^{\min}, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min}\Big]$, $\forall n, t$.

Proof. See Appendix D. □

Consider now the linear mapping

$$C_n^t = \frac{\hat{\lambda}_n^t}{\mu} + \frac{\overline{\alpha}_b}{\mu} + C_n^{\min} - P_{b,n}^{\min}, \quad \forall n. \tag{2.36}$$

It can be readily seen from Lemma 5 that $C_n^{\min} \leq C_n^t \leq C_n^{\max}$ holds for all n and t; i.e., (3.15c) are always satisfied under the proposed online scheme. With the battery (dis)charging dynamics (4.21b) naturally performed and the ramping constraint (2.23f) taken into account by the online decision, the feasibility of the control actions $\mathbf{x}(\hat{\boldsymbol{\pi}}_t)$ can be maintained for the original problem, provided that we select a stepsize $\mu \geq \underline{\mu}$.

2.3.3 Main theorem

Based on Lemmas 7 and 9, we are able to reach the following main result.

Theorem 1. *Upon setting $\hat{\lambda}_n^0 = \mu C_n^0 - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min}$, $\forall n$, and selecting a stepsize $\mu \geq \underline{\mu}$, the proposed algorithm yields a feasible dynamic control scheme for (2.27), which is near-optimal in the sense that*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right] \leq \Psi^* + \delta(\rho) + \mu M$$

where M, $\underline{\mu}$ and $\delta(\rho)$ are specified by Lemmas 3 and 5.

Clearly, the minimum optimality gap between Algorithm 6 and the offline scheduling is given by

$$\delta(\rho)+\underline{\mu}M = (1-\rho)P_c^{\max}\max\{\overline{\alpha}_b,\overline{\alpha}_c\} + \frac{1}{2}\underline{\mu}\left[J(1+\eta^2) + \textstyle\sum_{n=1}^{N}(\max\{P_{b,n}^{\max}, -P_{b,n}^{\min}\})^2\right].$$

The asymptotically optimal solution can be attained as $\rho \to 1$ (meaning that the ramping constraints are loose), and $\underline{\mu}$ is very small when the maximum difference between buying and selling prices $(\overline{\alpha}_b - \underline{\alpha}_s)$ approaches zero, or, the battery capacities $\{C_n^{\max}\}_n$ are very large. This makes sense intuitively because as $(\overline{\alpha}_b - \underline{\alpha}_s)$ approaches zero, purchasing extra power to charge the batteries will always make profit, and when batteries have large capacity, the upper bounds in (4.21b) are not in effect. In these cases, with a proper initialization, the proposed online policy using any μ will be feasible for (5.7), and the optimal Ψ^* will be reached as close as possible.

Remark 1. Readers familiar with optimization based on Lyapunov functions can recognize similarities between the stochastic dual sub-gradient based solver proposed here, and the Lyapunov optimization tools in [42, 72]. However, there are differences between two methods that 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 [42, 72] corresponds to the inverse of the stepsize μ in the stochastic optimization setup. In contrast, "virtual queues" are naturally emerging as Lagrange multiplier iterations in our stochastic optimization setup.

D2) Leveraging duality and online signal processing techniques, the stochastic dual subgradient iteration is also easy to interpret. The Lagrange multiplier for instance, can be viewed as the instantaneous charging price, which reveals the intuition behind real-time (dis)charging decisions, as discussed after Lemma 4. Weak duality is also utilized to prove Lemma 3. Finally, the dual subgradient iteration permeates results established for the least mean-square (LMS) algorithm - arguably the "workhorse" of adaptive schemes - to the problem at hand; e.g., LMS with constant stepsize only converges to the optimal Lagrange multiplier in the mean [69]. Thus, a large stepsize will lead to severe hovering around the equilibrium point, and thus it will incur considerable loss of optimality.

Table 2.1: Power supply parameters

$P_c^{\max}$	$C_n^{\min}$	$C_n^{\max}$	C_n^0	$P_{b,n}^{\min}$	$P_{b,n}^{\max}$	ρ
50	5	50	5	-5	5	0.6

Table 2.2: Data center cooling and operating parameters

D_{IT}	κ_1	κ_2	η	MP_j	J	I	T_{RA}	T_{OA}
100	10^{-10}	0.2	0.2	15	5	3	$35°C$	$20°C$

2.4 Numerical Evaluation

In this section, simulated tests are presented to demonstrate the merits of the proposed approach and justify the analytical claims of Section 2.3.

2.4.1 Experiment setup

The Matlab-based modeling package `CVX 2.1` [5] and the solver `SDPT3` [71] are used to solve the optimization problems involved. The considered system includes one data center, one conventional generator, one renewable generator, and $N = 2$ distributed energy storage units. The power supply limits and the corresponding parameters are listed in Table 4.1. The data center operating limits and the cooling parameters are listed in Table 3.2. Each type-i 'must-serve' workload and class-j delay-tolerant workload arrive according to a Poisson process with average IT demand 10 kWh/slot and 5 kWh/slot, respectively.

Two cases are considered for the energy market prices and the available renewables. In Case A (i.i.d. case), the purchase price α_b^t is uniformly distributed within $[50, 100]$ \$/MWh, and samples of the renewable supply $\{r^t\}$ are generated from a Weibull distributed wind speed and a wind-speed-to-wind-power mapping with maximum capacity $r^{\max} = 24$ kWh [89].

In Case B (real-data case), the purchase prices α_b^t are re-scaled from the day-ahead hourly electricity prices to the large general services in New York during Jan. 01–30, 2015 [7], while the renewable supply $\{r^t\}$ is a re-scaled version of the real-time hourly wind generation connected to the PJM grids at the same period [8]. The trend of energy purchase prices α_b^t and renewable supply

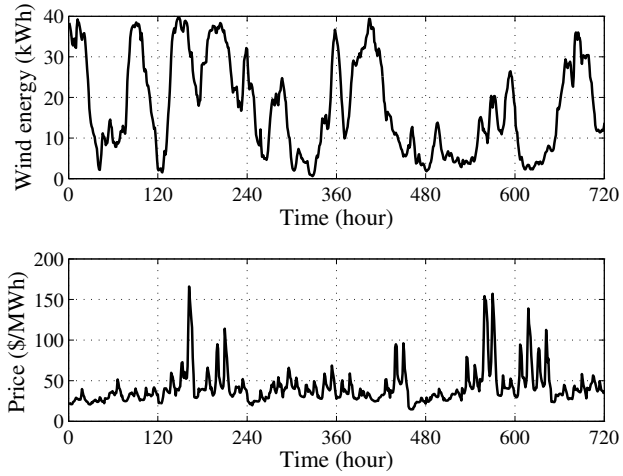

Figure 2.1: Hourly real-time wind power generation connected to PJM grids during Jan. 01–30, 2015 [8]; and day-ahead electricity prices in New York during Jan. 01–30, 2015 [7].

$\{r^t\}$ is shown in Fig. 2.1. Note that energy market prices and renewable energy generation here are highly correlated over time. While our performance analysis is carried out for the i.i.d. case, the proposed algorithm readily applies to this non-i.i.d. setup.

For both cases, the selling price is set to $\alpha_s^t = \xi\alpha_b^t$ with $\xi = 0.8$, and the CG generation cost is set to the average market price $(1/T)\sum_{t=1}^{T}\alpha_b^t$. Finally, slot duration is an hour with the entire time-horizon equal to 30 days (i.e., 720 slots), and the stepsize is chosen as $\mu \equiv \underline{\mu}$ [cf. Theorem 2] by default.

2.4.2 Benchmarks

To benchmark performance of the proposed algorithm, four baseline schemes are tested.

1) **ALG 1** (Renewable-aware, no cooling optimization, two-way trade, workload scheduling, with storage): ALG 1 is similar to the proposed algorithm except that no cooling optimization is performed.

2) **ALG 2** (Renewable-aware, no cooling optimization, two-way trade, no workload scheduling, with storage): ALG 2 is based on the approach in [72], where renewable energy is taken into account, but neither cooling optimization nor workload scheduling is carried out.

3) **ALG 3** (Renewable-oblivious, no cooling optimization, two-way trade, no workload schedul-

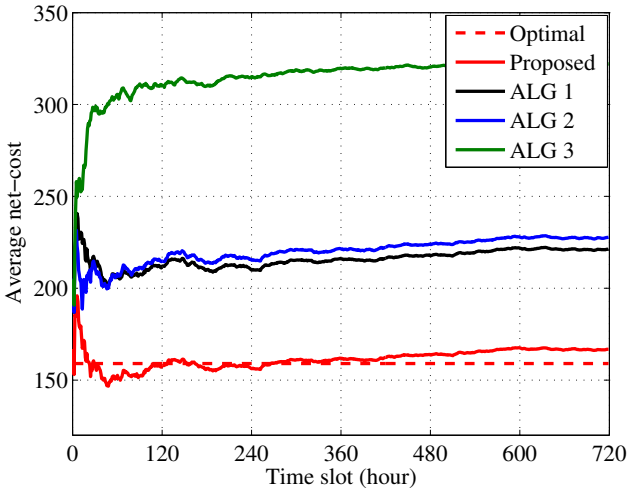

Figure 2.2: Comparison of average net-costs.

ing, without storage): ALG 4 is widely used in practice to minimize only the energy transaction cost without any consideration on workload management, renewable energy, cooling optimization or storage.

4) **Optimal**: Assuming all needed statistics of randomness are known a-priori, the offline optimal algorithm is also introduced to solve (5.7) over the entire horizon $T = 720$ slots. This optimal algorithm cannot work in practice due to the lack of future information.

Note that [72] does not account for real-time two-way energy transaction, workload management, and cooling optimization. For fair comparison, chilled-water cooling is utilized to calculate the final net-cost for ALGs 1-3, while two-way energy transaction is also allowed.

2.4.3 Case A (i.i.d. data)

In Fig. 4.4, the proposed Algorithm 1 is compared with ALGs 1-3, and also against the offline optimal benchmark, in terms of the average net-cost. Within 720 iterations (time slots) the proposed algorithm converges to a much lower net-cost than ALGs 1-3. The net-costs of ALGs 1-3 are about $33\%, 37\%$ and 95% larger than that of the proposed algorithm. Intuitively speaking, this is because the proposed algorithm takes both cooling optimization and workload management into account. It also leverages the renewable energy and energy storage units to hedge against future fluctuation of workload demands and energy prices. These advantages cannot be fully exploited by ALGs 1-3.

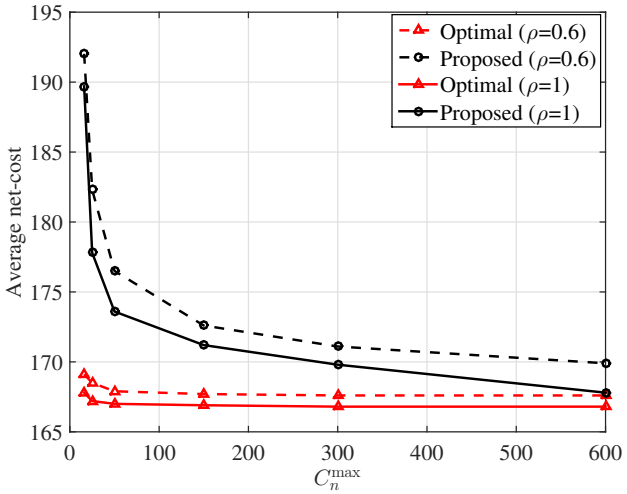

Figure 2.3: Average net-cost versus $C_n^{\max}$ and ρ.

On the other hand, without any future information, the proposed online algorithm incurs only 5% optimality loss compared with the offline optimal approach.

Fig. 2.3 demonstrates the impact of battery capacity $C_n^{\max}$ and ramping parameter ρ on algorithm performance. For a fixed $C_n^{\max}$, a larger ρ results in a smaller average net-cost and a smaller optimality gap. This is consistent with Lemma 7 and also intuitive since a larger ρ implies a looser ramping constraint, which endows the proposed algorithm with more freedom to purchase cheaper energy from CG. For a fixed ρ, the optimality gap decreases as $C_n^{\max}$ increases, as a larger $C_n^{\max}$ allows the algorithm to choose a smaller stepsize $\underline{\mu}$ [cf. Lemma 9].

To further delineate the trade off between the battery feasibility and the algorithm optimality, Figs. 4.7-4.8 depict the average net-cost and the battery SoC evolution for different stepsizes μ. With the same parameters, the proposed algorithm converges faster with a larger stepsize μ (i.e., $\mu = \underline{\mu}$), but incurs lower net-cost with a small stepsize μ (i.e., $\mu = 0.1\underline{\mu}$). This is precisely consistent with Lemma 7 in the sense that the optimality gap is proportional to the stepsize μ. However, recall that arbitrarily small stepsize μ may affect feasibility of the proposed online scheme [cf. Lemma 9]. In Fig. 4.8, it turns out that the SoC is always feasible ($C_1^{\min} \leq C_1^t \leq C_1^{\max}$) when $\mu = \underline{\mu}$. In contrast, if the stepsize is selected as $\mu = 0.1\underline{\mu}$, which does not satisfy the stepsize condition in Lemma 9, then C_1^t exceeds its physical upper bound immediately.

The evolution of energy purchase prices α_b^t, selling prices α_s^t, Lagrange multipliers $\hat{\lambda}_1^t$, as well

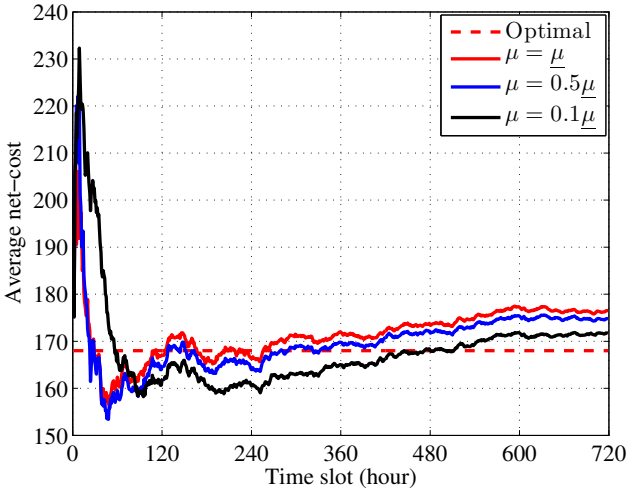

Figure 2.4: Average net-cost versus μ.

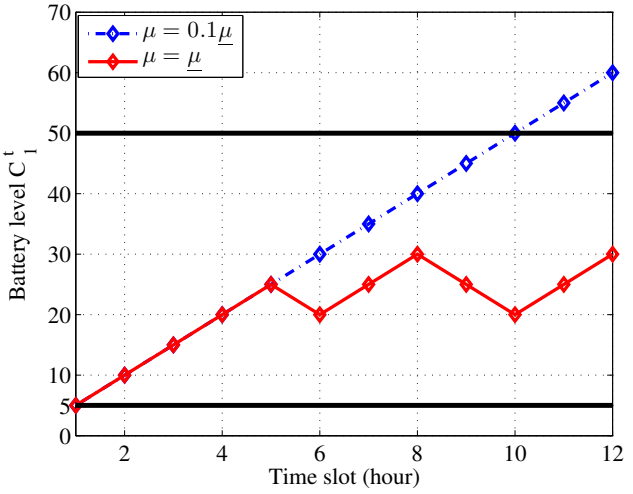

Figure 2.5: The battery state-of-charge C_1^t versus stepsize μ.

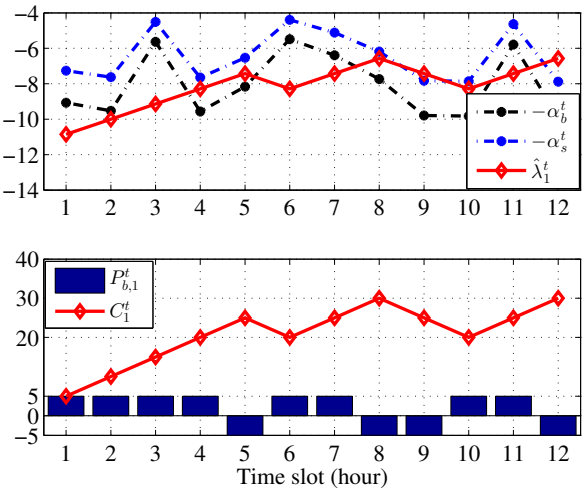

Figure 2.6: Schedule of battery power $P_{b,1}^t$.

as the real-time battery SoC C_1^t and battery (dis)charging amount $P_{b,1}^t$ are shown in Fig. 4.5. It can be seen that $P_{b,1}^t = P_{b,1}^{\min}$ when $\hat{\lambda}_1^t > -\alpha_s^t$ at $t = 9, 12$, while $P_{b,1}^t = P_{b,1}^{\max}$ when $\hat{\lambda}_1^t < -\alpha_b^t$ at $t = 1, 2, 3, 6, 7, 11$. Notice that when $\hat{\lambda}_1^t \in [-\alpha_b^t, -\alpha_s^t]$ at $t = 4, 5, 8, 10$, one must resort to solving (2.34) numerically to obtain $P_{b,1}^t$, since the sufficient conditions for (dis)charging actions $P_{b,n}^t$ in Lemma 8 are not satisfied. Clearly, the Lagrange multiplier $\hat{\lambda}_1^t$ is in fact a mapping of the real-time battery SoC C_1^t [cf. (3.36)]. Such mapping relationships are also true for the slots $t > 12$ and $P_{b,2}^t$.

The long-term QoS ratio [cf. (2.4)] of the proposed algorithm is depicted in Fig. 4.9, where the QoS ratio of the proposed algorithm quickly converges to the threshold $\eta = 0.2$ as the number of iterations increases. This corroborates our assertion that time-average constraints (4.21c) are asymptotically satisfied by leveraging the stochastic subgradient strategy [50].

2.4.4 Case B (real-data)

Fig. 2.8 compares the average net-cost and IT consumption [cf. (4.6)] of the proposed algorithm and ALGs 1-3. It can be seen that the proposed algorithm reduces the net-cost by 15%-47%, while all algorithms have similar average IT consumption. The result is expected since the proposed algorithm optimizes the cooling efficiency and intelligently schedules IT workloads according to current energy prices and task revenues. In contrast, ALG 1 ignores cooling consumption and

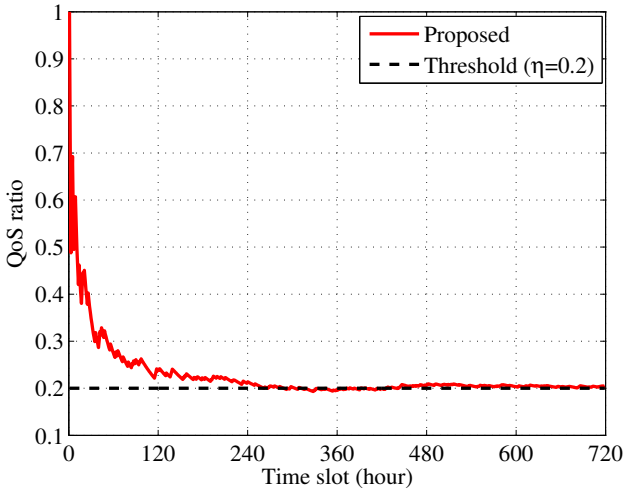

Figure 2.7: QoS ratio of delay-tolerant workloads in class 1.

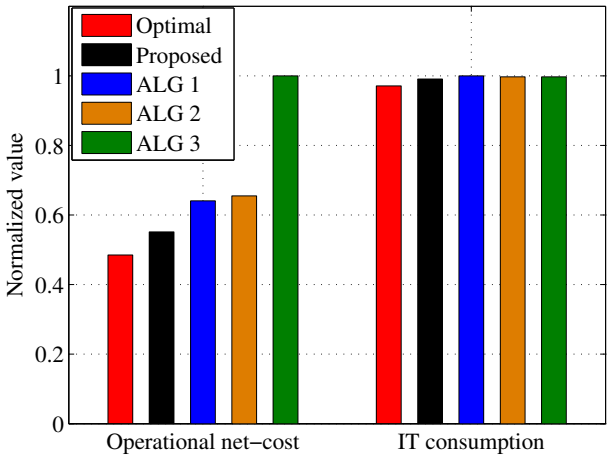

Figure 2.8: Comparison of average net-cost and IT consumption.

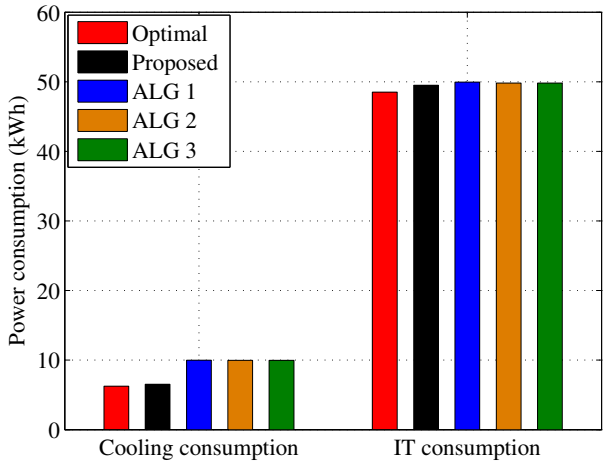

Figure 2.9: Comparison of cooling and IT consumptions.

thus underestimates the total power demand, which results in accommodating more delay-tolerant workloads than the proposed algorithm. ALG 2 incurs a higher net-cost since it does not consider cooling consumption and workload management, whereas ALG 3 is oblivious to not only cooling consumption and workload management but also renewable energy and storage units. At the same time, the proposed algorithm only exhibits 14% optimality loss, compared with the ideally optimal algorithm having all future information available. Note that smaller optimality loss can be expected when larger batteries are deployed in this setup [cf. Fig. 2.3].

The average cooling energy consumption and IT revenue are compared with the average IT consumption in Figs. 2.9 and 2.10, separately. Clearly, the proposed algorithm reduces the cooling energy consumption by almost 35%, while it has only 1% less IT consumption than ALGs 1-3. Further, it is shown that by using combined cooling sources, the average cooling coefficient of the proposed algorithm is around 0.13, which is more efficient than simple chilled-water cooling with a constant coefficient 0.2. This result is of interest and meaningful. It implies that by integrating cooling optimization with workload management, the proposed algorithm can use less energy to serve the same amount of IT consumption. Furthermore, Fig. 2.10 shows that by incorporating workload management, the proposed algorithm can earn 5% more IT revenue with the same IT consumption than other algorithms without workload management.

Fig. 2.11 depicts the average power schedule of the proposed algorithm over a 24-hour period,

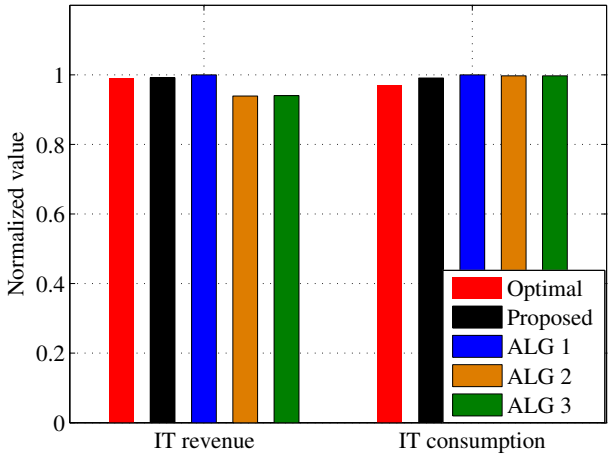

Figure 2.10: Comparison of IT revenue and IT consumption.

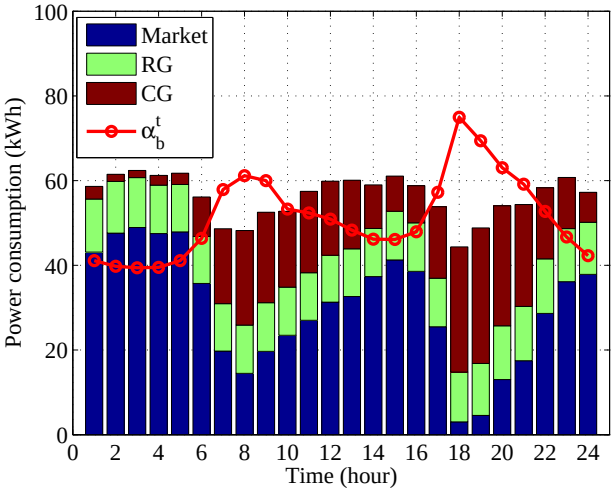

Figure 2.11: Power schedule of the proposed algorithm.

and the trend of energy purchase prices α_b^t is also shown to illustrate the resultant online policy. One observation is that the hourly power consumption closely reflects the instantaneous energy purchase price α_b^t. Specifically, the proposed method tends to consume more power when α_b^t is lower (24PM to 5AM), and less power when α_b^t is higher (7AM to 10AM, and 17PM to 21PM). Moreover, the lower energy purchase price α_b^t in the proposed method encourages purchasing more energy from the external grid market, while the peak of α_b^t results in a higher power usage from the CG.

2.5 Appendices

2.5.1 Proof of Lemma 2

Let $\hat{\mathbf{x}}_t := \{\hat{\mathbf{w}}^t, \hat{\mathbf{p}}_b^t, \hat{P}_c^t\}$ denote the optimal solution for (2.34), and $\check{\mathbf{x}}_t := \{\check{\mathbf{w}}^t, \check{\mathbf{p}}_b^t, \check{P}_c^t\}$ the optimal solution for (2.35). Construct a vector $\mathring{\mathbf{x}}_t := \{\check{\mathbf{w}}^t, \check{\mathbf{p}}_b^t, \mathring{P}_c^t\}$. Note that the ramping constraint $|P_c^t - \hat{P}_c^{t-1}| \leq \rho P_c^{\max}$ in (2.34) is only relevant to P_c^t. Recall that $\{\check{\mathbf{w}}^t, \check{\mathbf{p}}_b^t, \check{P}_c^t\}$ satisfies the constraints (3.15d)-(3.15e) and (4.21d)-(4.21e). Upon selecting any $\mathring{P}_c^t \in [\hat{P}_c^{t-1} - \rho P_c^{\max}, \hat{P}_c^{t-1} + \rho P_c^{\max}]$, $\mathring{\mathbf{x}}_t$ will be in the feasible set of (2.34). Let $\hat{\Omega}(\mathring{\mathbf{x}}_t)$ denote the value of objective function for the feasible solution $\mathring{\mathbf{x}}_t$. It clearly holds that $\hat{\Omega}^* - \check{\Omega}^* \leq \hat{\Omega}(\mathring{\mathbf{x}}_t) - \check{\Omega}^*$, since $\mathring{\mathbf{x}}_t$ is a feasible solution but not necessarily the minimizer of (2.34). As a consequence, we deduce that [cf. definitions of δ_-^t and δ_+^t in Lemma 1]

$$\begin{aligned}
\hat{\Omega}^* - \check{\Omega}^* &\leq \hat{\Omega}(\mathring{\mathbf{x}}_t) - \check{\Omega}^* = \Psi(\mathring{\mathbf{x}}_t) - \Psi(\check{\mathbf{x}}_t) \\
&= \delta_-^t \left| \check{d}^t + c(\check{d}^t) + \textstyle\sum_n \check{P}_{b,n}^t - \mathring{P}_c^t - r^t \right| + \alpha_c^t \mathring{P}_c^t \\
&\quad - \delta_-^t \left| \check{d}^t + c(\check{d}^t) + \textstyle\sum_n \check{P}_{b,n}^t - \check{P}_c^t - r^t \right| - \alpha_c^t \check{P}_c^t \\
&\quad + \delta_+^t (\check{P}_c^t - \mathring{P}_c^t) \\
&\leq \delta_-^t \left| \check{P}_c^t - \mathring{P}_c^t \right| + \alpha_c^t (\mathring{P}_c^t - \check{P}_c^t) + \delta_+^t (\check{P}_c^t - \mathring{P}_c^t)
\end{aligned}$$

where the last inequality follows from the triangle inequality.

Consider the next three cases.

c1) If $\check{P}_c^t \in [\hat{P}_c^{t-1} - \rho P_c^{\max}, \hat{P}_c^{t-1} + \rho P_c^{\max}]$, then simply let $\mathring{P}_c^t = \check{P}_c^t$ (i.e., $\mathring{\mathbf{x}}_t = \check{\mathbf{x}}_t$). It is then clear that $\hat{\Omega}^* - \check{\Omega}^* = \hat{\Omega}(\mathring{\mathbf{x}}_t) - \check{\Omega}^* = 0$.

c2) If $\check{P}_c^t > \hat{P}_c^{t-1} + \rho P_c^{\max}$, then pick $\mathring{P}_c^t = \hat{P}_c^{t-1} + \rho P_c^{\max}$ in $\mathring{\mathbf{x}}_t$ to arrive at

$$\begin{aligned}
&\delta_-^t \left| \check{P}_c^t - \mathring{P}_c^t \right| + \alpha_c^t (\mathring{P}_c^t - \check{P}_c^t) + \delta_+^t (\check{P}_c^t - \mathring{P}_c^t) \\
&\quad = \alpha_b^t (\check{P}_c^t - \mathring{P}_c^t) + \alpha_c^t (\mathring{P}_c^t - \check{P}_c^t) \\
&\quad = (\alpha_b^t - \alpha_c^t)(\check{P}_c^t - \hat{P}_c^{t-1} - \rho P_c^{\max}) \\
&\quad \leq \alpha_b^t (\check{P}_c^t - \rho P_c^{\max}) \\
&\quad \leq \overline{\alpha}_b (1-\rho) P_c^{\max}
\end{aligned}$$

where the last equality holds because $\check{P}_c^t \leq P_c^{\max}$.

c3) If $\check{P}_c^t < \hat{P}_c^{t-1} - \rho P_c^{\max}$, then select $\mathring{P}_c^t = \hat{P}_c^{t-1} - \rho P_c^{\max}$ in $\mathring{\mathbf{x}}_t$. Similarly, we have

$$\begin{aligned}
&\delta_-^t \left| \check{P}_c^t - \mathring{P}_c^t \right| + \delta_+^t (\check{P}_c^t - \mathring{P}_c^t) + \alpha_c^t (\mathring{P}_c^t - \check{P}_c^t) \\
&\quad = \alpha_c^t (\mathring{P}_c^t - \check{P}_c^t) - \alpha_s^t (\mathring{P}_c^t - \check{P}_c^t) \\
&\quad = (\alpha_c^t - \alpha_s^t)(\hat{P}_c^{t-1} - \rho P_c^{\max} - \check{P}_c^t) \\
&\quad \leq \alpha_c^t (\hat{P}_c^{t-1} - \rho P_c^{\max}) \\
&\quad \leq \overline{\alpha}_c (1-\rho) P_c^{\max}
\end{aligned}$$

where the last equality is due to $\hat{P}_c^{t-1} \leq P_c^{\max}$.

Combining cases c1) – c3), it readily follows that $\hat{\Omega}^* - \check{\Omega}^* \leq (1-\rho) P_c^{\max} \max\{\overline{\alpha}_b, \overline{\alpha}_c\}$.

2.5.2 Proof of Lemma 3

Squaring the update in (2.33a) yields

$$\begin{aligned}
\|\hat{\boldsymbol{\lambda}}_{t+1}\|_2^2 &= \|\hat{\boldsymbol{\lambda}}_t\|_2^2 + 2\mu \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t) + \mu^2 \|\mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t)\|_2^2 \\
&\leq \|\hat{\boldsymbol{\lambda}}_t\|_2^2 + 2\mu \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t) + \mu^2 \textstyle\sum_{n=1}^N (\max\{P_{b,n}^{\max}, -P_{b,n}^{\min}\})^2
\end{aligned}$$

where the last inequality follows from constraints (3.15d).

Likewise, squaring the update in (2.33b) implies [cf. the definition of $\oslash$ in Section I]

$$\|\hat{\boldsymbol{\nu}}_{t+1}\|_2^2 \leq \|\hat{\boldsymbol{\nu}}_t\|_2^2 + 2\mu \hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \check{\mathbf{w}}^t\right) + J\mu^2(1+\eta^2)$$

which leads to

$$\frac{1}{2}\|\hat{\boldsymbol{\pi}}_{t+1}\|_2^2 - \frac{1}{2}\|\hat{\boldsymbol{\pi}}_t\|_2^2 \leq \mu\hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \breve{\mathbf{w}}^t\right) + \mu\hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t) + \mu^2 M. \tag{2.37}$$

Upon adding $\mu\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)$ [cf. (2.34)], and taking expectations on both sides of (2.37), we find

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}\left[\|\hat{\boldsymbol{\pi}}_{t+1}\|_2^2 - \|\hat{\boldsymbol{\pi}}_t\|_2^2\right] + \mu\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right] \\
\leq &\mu\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right) + \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t) + \hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \breve{\mathbf{w}}^t\right)\right] + \mu^2 M.
\end{aligned}$$

Summing both sides of the last inequality over t and dividing both sides by μT, we arrive at

$$\begin{aligned}
&\frac{1}{2\mu T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\hat{\boldsymbol{\pi}}_{t+1}\|_2^2 - \|\hat{\boldsymbol{\pi}}_t\|_2^2\right] + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right] \\
= &\frac{1}{2\mu T}\mathbb{E}\left[\|\hat{\boldsymbol{\pi}}_T\|_2^2 - \|\hat{\boldsymbol{\pi}}_0\|_2^2\right] + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right] \\
\leq &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right) + \hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \breve{\mathbf{w}}^t\right) + \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t)\right] + \mu M
\end{aligned}$$

from which letting $T \to \infty$ yields

$$\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right)\right] \\
\leq &\,\mu M - \lim_{T\to\infty}\frac{1}{2\mu T}\mathbb{E}\left[\|\hat{\boldsymbol{\pi}}_T\|_2^2\right] + \lim_{T\to\infty}\frac{1}{2\mu T}\mathbb{E}\left[\|\hat{\boldsymbol{\pi}}_0\|_2^2\right] + \\
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right) + \hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \breve{\mathbf{w}}^t\right) + \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t)\right] \\
\leq &\,\mu M + \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\hat{\boldsymbol{\pi}}_t)\right) + \hat{\boldsymbol{\lambda}}_t^\top \mathbf{p}_b^t(\hat{\boldsymbol{\pi}}_t) + \hat{\boldsymbol{\nu}}_t^\top \left(1 - \eta - \mathbf{w}^t(\hat{\boldsymbol{\pi}}_t) \oslash \breve{\mathbf{w}}^t\right)\right] \\
= &\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\hat{\Omega}^*(\hat{\boldsymbol{\pi}}_t)\right] + \mu M \\
\overset{(a)}{\leq} &\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\breve{\Omega}^*(\hat{\boldsymbol{\pi}}_t)\right] + \delta(\rho) + \mu M \\
\overset{(b)}{=} &\,\mathcal{L}(\mathbf{x}(\hat{\boldsymbol{\pi}}_t), \hat{\boldsymbol{\pi}}_t) + \delta(\rho) + \mu M \overset{(c)}{=} \mathcal{D}(\hat{\boldsymbol{\pi}}_t) + \delta(\rho) + \mu M \\
\overset{(d)}{\leq} &\,\overline{\Psi}^* + \delta(\rho) + \mu M \overset{(e)}{\leq} \Psi^* + \delta(\rho) + \mu M
\end{aligned}$$

where inequality (a) follows from Lemma 2; equality (b) follows from the definition of the Lagrangian in (5.11) with $\mathbf{x}(\hat{\boldsymbol{\pi}}_t)$ denoting the optimal primal variables given by (2.34); equality (c) comes from the definition of the dual function; inequality (d) follows from the weak duality [cf. (2.27a)]; and inequality (e) holds since (2.27) is a relaxation of (5.7).

2.5.3 Proof of Lemma 4

Algorithm 6 solves the real-time problem (2.34) per slot t. In particular, $\{P_{b,n}^t(\hat{\boldsymbol{\pi}}_t)\}$ are obtained by solving [cf. (2.22)]

$$\begin{aligned}\min_{\mathbf{p}_b^t}\ & \delta_-^t|P_{\text{out}}^t - P_{\text{in}}^t| + \delta_+^t(P_{\text{out}}^t - P_{\text{in}}^t) + \sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t \\ \text{s. t. }\ & P_{b,n}^{\min} \le P_{b,n}^t \le P_{b,n}^{\max},\ \forall n.\end{aligned}$$

Consider the following two cases.

i) If $P_{\text{out}}^t \ge P_{\text{in}}^t$, then

$$\begin{aligned}&\delta_-^t|P_{\text{out}}^t - P_{\text{in}}^t| + \delta_+^t(P_{\text{out}}^t - P_{\text{in}}^t) + \textstyle\sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t \\ &= \alpha_b^t\Big[d^t + f(d^t) + \textstyle\sum_{n=1}^N P_{b,n}^t - P_c^t - r^t\Big] + \textstyle\sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t \\ &= \alpha_b^t\left[d^t + f(d^t) - P_c^t - r^t\right] + \textstyle\sum_{n=1}^N (\hat{\lambda}_n^t + \alpha_b^t) P_{b,n}^t.\end{aligned}$$

It is easy to see that

$$P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = \begin{cases} P_{b,n}^{\min}, & \text{if } \hat{\lambda}_n^t + \alpha_b^t > 0 \\ P_{b,n}^{\max}, & \text{if } \hat{\lambda}_n^t + \alpha_b^t < 0. \end{cases}$$

ii) If $P_{\text{out}}^t < P_{\text{in}}^t$, then

$$\begin{aligned}&\delta_-^t|P_{\text{out}}^t - P_{\text{in}}^t| + \delta_+^t(P_{\text{out}}^t - P_{\text{in}}^t) + \textstyle\sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t \\ &= \alpha_s^t\Big[d^t + f(d^t) + \textstyle\sum_{n=1}^N P_{b,n}^t - P_c^t - r^t\Big] + \textstyle\sum_{n=1}^N \hat{\lambda}_n^t P_{b,n}^t \\ &= \alpha_s^t\left[d^t + f(d^t) - P_c^t - r^t\right] + \textstyle\sum_{n=1}^N (\hat{\lambda}_n^t + \alpha_s^t) P_{b,n}^t.\end{aligned}$$

Similarly, it holds that

$$P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = \begin{cases} P_{b,n}^{\min}, & \text{if } \hat{\lambda}_n^t + \alpha_s^t > 0 \\ P_{b,n}^{\max}, & \text{if } \hat{\lambda}_n^t + \alpha_s^t < 0. \end{cases}$$

Combining cases i) and ii), one deduces that if per slot t, $\hat{\lambda}_n^t > \max\{-\alpha_b^t, -\alpha_s^t\} = -\alpha_s^t$, then $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = P_{b,n}^{\min}$. Likewise, if $\hat{\lambda}_n^t < \min\{-\alpha_b^t, -\alpha_s^t\} = -\alpha_b^t$, then $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = P_{b,n}^{\max}$, and the lemma follows readily.

2.5.4 Proof of Lemma 5

The argument proceeds by induction. First, set $\hat{\lambda}_n^0 \in \Big[-\overline{\alpha}_b + \mu P_{b,n}^{\min}, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min} \Big], \forall n$, and suppose that this holds for $\hat{\lambda}_n^t$. We will show that the bounds hold for $\hat{\lambda}_n^{t+1}$, as well as for subsequent instances. Consider the following three cases.

c1) If $\hat{\lambda}_n^t \in (-\underline{\alpha}_s, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min}]$, then it follows from Lemma 8 that $\hat{\lambda}_n^{t+1} = \hat{\lambda}_n^t + \mu P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) \in (-\overline{\alpha}_b + \mu P_{b,n}^{\min}, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min})$ holds considering the facts $-\overline{\alpha}_b < -\underline{\alpha}_s$ and $P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) = P_{b,n}^{\min}$.

c2) If $\hat{\lambda}_n^t \in [-\overline{\alpha}_b, -\underline{\alpha}_s]$, then $\hat{\lambda}_n^{t+1} = \hat{\lambda}_n^t + \mu P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) \in [-\overline{\alpha}_b + \mu P_{b,n}^{\min}, -\underline{\alpha}_s + \mu P_{b,n}^{\max}] \subseteq [-\overline{\alpha}_b + \mu P_{b,n}^{\min}, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min}]$, since $\mu \geq \underline{\mu}$.

c3) If $\hat{\lambda}_n^t \in [-\overline{\alpha}_b + \mu P_{b,n}^{\min}, -\overline{\alpha}_b)$, then Lemma 8 implies that $\hat{\lambda}_n^{t+1} = \hat{\lambda}_n^t + \mu P_{b,n}^t(\hat{\boldsymbol{\pi}}_t) \in [-\overline{\alpha}_b + \mu P_{b,n}^{\min} + \mu P_{b,n}^{\max}, -\overline{\alpha}_b + \mu P_{b,n}^{\max}) \subseteq [-\overline{\alpha}_b + \mu P_{b,n}^{\min} + \mu P_{b,n}^{\max}, -\underline{\alpha}_s + \mu P_{b,n}^{\max}) \subseteq (-\overline{\alpha}_b + \mu P_{b,n}^{\min}, \mu C_n^{\max} - \mu C_n^{\min} - \overline{\alpha}_b + \mu P_{b,n}^{\min})$, where the last step follows because $P_{b,n}^{\max} > 0$, $-\overline{\alpha}_b < -\underline{\alpha}_s$, and the fact under c2).

Chapter 3

Distributed Stochastic Geographical Load Balancing with Incentive Payment

3.1 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 $\mathcal{J} := \{1, 2, \cdots, J\}$ MNs, and $\mathcal{I} := \{1, 2, \cdots, I\}$ 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 subsections 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.

3.1.1 Traffic workloads and network constraints

Suppose that each MN collects two types of workloads: delay-sensitive interactive and delay-tolerant workloads [85]. 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.

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

$$\textstyle\sum_{i\in\mathcal{I}_j} v_{i,j,t} = V_{j,t}, \quad \forall j,t. \tag{3.1}$$

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 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 unserved workload is buffered in queues (one per class of DW) obeying the following dynamic recursion

$$Y^{\mathrm{mn}}_{j,q,t+1} = \left[Y^{\mathrm{mn}}_{j,q,t} + W_{j,q,t} - \textstyle\sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t} \right]^{\infty}_{0}, \ \forall j,q,t, \tag{3.2}$$

where $Y^{\mathrm{mn}}_{j,q,t}$ is the queue length of DW q in MN j at the beginning of slot t.

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

$$Y^{\mathrm{dc}}_{i,q,t+1} = \left[Y^{\mathrm{dc}}_{i,q,t} - w_{i,q,t} + \textstyle\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} \right]^{\infty}_{0}, \ \forall i,q,t, \tag{3.3}$$

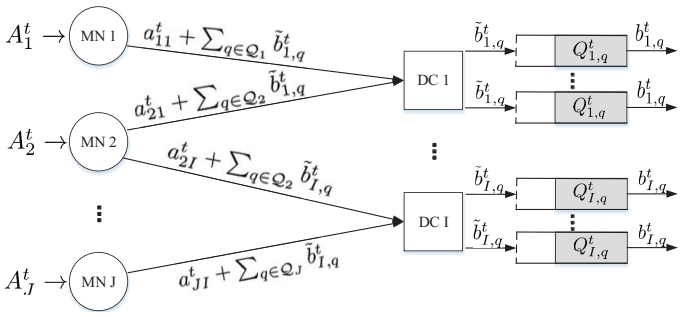

Figure 3.1: A geographical load balancing system diagram.

where $Y_{i,q,t}^{\mathrm{dc}}$ is the queue length of DW q in DC i at the beginning of slot t. Queue dynamics slightly different from the one in (5.1)-(5.2) can also be considered [33, 85], but such differences are not relevant for the subsequent analysis.

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

$$d_{i,t} = \sum_{j\in\mathcal{J}} v_{i,j,t} + \sum_{q\in\mathcal{Q}} w_{i,q,t}, \quad \forall i, t. \tag{3.4}$$

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

$$v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t} \leq B_{j,i}, \quad \forall j, i, t, \tag{3.5}$$

with $B_{j,i} = 0$ if MN j and DC i are not connected. The workload allocation diagram is sketched in Fig. 3.1.

3.1.2 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_i^{it}, the power consumed by a single server in DC i. With $c \in [0, 1]$ denoting the speed of server or, alternatively, the CPU usage (processed work divided by the server capacity), and $\overline{P}_i^{\mathrm{s}}$ the peak power consumption of a server in DC i, then P_i^{it} can be approximated as $P_i^{\mathrm{it}}(c) = \overline{P}_i^{\mathrm{s}}\,(\varrho c^{\sigma} + 1 - \varrho)$, 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 $\sigma \geq 1$ is typically set as 2 in state-of-the-art servers [78]. Assume servers in DC i are all identical, and let M_i denote the total number of servers in DC i. Given the total IT demand $d_{i,t}$ in DC i at time t, it follows from the convexity of $P_i^{\mathrm{it}}(c)$ that the most energy-efficient allocation is to allocate $d_{i,t}$ uniformly across servers. In this way, with D_i denoting the capacity of each server at DC i, each server is running at a speed $(d_{i,t}/M_i)/D_i \in [0,1]$, and the total power consumption can be calculated as

$$P_i^{\mathrm{it}}(d_{i,t}) = \frac{\varrho d_{i,t}^2}{M_i D_i^2}\overline{P}_i^{\mathrm{s}} + (1-\varrho)M_i\overline{P}_i^{\mathrm{s}}. \tag{3.6}$$

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 promoted. Additional details as well as specific research on dynamic sizing of DCs can be found in [46].

Along with the increasing density of IT equipment in DCs, a considerable amount of electricity is consumed by the cooling system [9]. 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 [48]. 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., [21, 48].

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

$$0 \leq P_{i,t}^{\mathrm{g}} \leq \overline{P}_i^{\mathrm{g}}, \ \ \forall t. \tag{3.7}$$

- $P^{\mathrm{r}}_{i,t}$ is the renewable energy generated at the beginning of slot t by the RG in DC i, which is upper-bounded too $P^{\mathrm{r}}_{i,t} \leq \overline{P}^{\mathrm{r}}_{i}, \forall t$.

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

$$\underline{Y}^{\mathrm{b}}_{i} \leq Y^{\mathrm{b}}_{i,t} \leq \overline{Y}^{\mathrm{b}}_{i}, \ \ \forall i, t \tag{3.8}$$

$$Y^{\mathrm{b}}_{i,t+1} = Y^{\mathrm{b}}_{i,t} + P^{\mathrm{b}}_{i,t}, \ \ \forall i, t \tag{3.9}$$

$$\underline{P}^{\mathrm{b}}_{i} \leq P^{\mathrm{b}}_{i,t} \leq \overline{P}^{\mathrm{b}}_{i}, \ \ \forall i, t \tag{3.10}$$

  where the bounds on the (dis)charging amount $\underline{P}^{\mathrm{b}}_{i} < 0$ and $\overline{P}^{\mathrm{b}}_{i} > 0$ in (4.17) 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, $P^{\mathrm{m}}_{i,t}$ denotes the energy that DC i buys from the *market* at time t. Since a two-way energy trading facility is considered, if negative, $P^{\mathrm{m}}_{i,t}$ 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

$$P^{\mathrm{m}}_{i,t} + P^{\mathrm{g}}_{i,t} + P^{\mathrm{r}}_{i,t} = P^{\mathrm{it}}_{i}(d_{i,t}) + P^{\mathrm{ac}}_{i}(d_{i,t}) + P^{\mathrm{b}}_{i,t}. \tag{3.11}$$

Under constraints (4.11)-(3.11), $P^{\mathrm{r}}_{i,t}$ is the state variable, while $\{P^{\mathrm{g}}_{i,t}, P^{\mathrm{b}}_{i,t}, P^{\mathrm{m}}_{i,t}\}$ are optimization variables.

3.1.3 Revenues and operation costs

Starting with the service and distribution of the workloads, we consider the *revenue* for IWs for MN j as $U^{\mathrm{v}}_{j}(\,\cdot\,)$, and the revenue of class-q DWs a strictly concave function $U^{\mathrm{w}}_{q}(\,\cdot\,)$. On the other hand, distribution of workloads across the network generates bandwidth *costs*. To this end, we will use the

convex function $G^{\mathrm{d}}_{i,j}(\,\cdot\,)$ to denote the cost for 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 $\alpha^{\mathrm{p}}_{i,t}$ (if $P^{\mathrm{m}}_{i,t} > 0$), or, sell energy to the markets at price $\alpha^{\mathrm{s}}_{i,t}$ if ($P^{\mathrm{m}}_{i,t} < 0$). Clearly, the shortage energy that needs to be purchased by the DC is $[P^{\mathrm{m}}_{i,t}]^{+}$; while the surplus energy that can be sold is $[P^{\mathrm{m}}_{i,t}]^{-}$. Notwithstanding, we shall always consider that $\alpha^{\mathrm{p}}_{i,t} \geq \alpha^{\mathrm{s}}_{i,t}$. This will prevent less relevant buy-and-sell activities of the DC for profit and will guarantee that, for any given time instant 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

$$G^{\mathrm{e}}_{i}(P^{\mathrm{m}}_{i,t}) := \alpha^{\mathrm{p}}_{i,t}[P^{\mathrm{m}}_{i,t}]^{+} - \alpha^{\mathrm{s}}_{i,t}[P^{\mathrm{m}}_{i,t}]^{-}. \tag{3.12}$$

Moreover, we will use the convex function $G^{\mathrm{c}}_{i}(\,\cdot\,)$ to denote the *cost* of CG during time t, which typically is smooth quadratic [80]. Finally, to model the potential battery degeneration during the charging/discharging cycle, a strongly convex (dis)charging cost $G^{\mathrm{b}}_{i}(\,\cdot\,)$ can be employed to prevent fast and frequent (dis)charging of batteries [62].

3.1.4 Incentive payment models

While most existing works (e.g., [21,46,48,82]) assume that IWs are fixed and inelastic, a number of interactive services tolerate their partial execution [83] (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 SLA [35]. These incentive prices are usually offered when the local marginal price (LMP) is high, or when the grid operator sends emergency demand response (EDR) signals such as a power outage [88].

Mathematically, let $p_{j,t}$ denote the incentive price that at time t MN j pays to users willing to reduce their interactive workload [49]. This way, if users reduce their demand by an amount $\check{V}_{j,t}$, the MN will pay $p_{j,t}\check{V}_{j,t}$. To model the users' reaction to the incentive price, we assume that users, when *reducing their workload demand by* an amount $\check{V}_{j,t}$, incur a strictly convex unsatisfactory cost $G^{\mathrm{u}}_{j}(\check{V}_{j,t}) = \kappa_j(\check{V}_{j,t})^2$, where the coefficient $\kappa_j > 0$ is learned from historical data. Hence, rational

users will set their demand by solving the following optimization problem

$$\max_{0\leq\check{V}_{j,t}\leq\eta V_{j,t}} p_{j,t}\check{V}_{j,t} - G_j^{\mathrm{u}}(\check{V}_{j,t}) \tag{3.13}$$

where $V_{j,t}$ is the total interactive workload demand for MN j without incentive payment [cf. (4.1)], and η is the threshold of maximum workload reduction.

Note that (3.13) admits a closed-form solution, namely $\check{V}_{j,t}(p_{j,t}) = (\nabla G_j^{\mathrm{u}})^{-1}(p_{j,t}) = [p_{j,t}/2\kappa_j]_0^{\eta V_{j,t}}$, 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 $\tilde{V}_{j,t} = V_{j,t} - \check{V}_{j,t}(p_{j,t})$, which can be written as a convex (linear) function of $p_{j,t}$ as $\tilde{V}_{j,t} = [V_{j,t} - p_{j,t}/(2\kappa_j)]_{(1-\eta)V_{j,t}}^{\infty}$.

3.2 Stochastic Load Balancing

Section 5.1 identified the variables, costs and constraints that must be accounted for 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 $\mathbf{x}_t := \{v_{i,j,t}, w_{i,q,t}, \tilde{w}_{i,j,q,t}, d_{i,t}, p_{j,t}, P_{i,t}^{\mathrm{m}}, P_{i,t}^{\mathrm{g}}, P_{i,t}^{\mathrm{b}},\ \forall i,j,q\}$. The resultant aggregated *network cost* for the considered MN-DC network *at time* t is

$$\begin{aligned}\Psi_t(\mathbf{x}_t) := &\sum_{i\in\mathcal{I}}\Big(G_i^{\mathrm{e}}(P_{i,t}^{\mathrm{m}}) + G_i^{\mathrm{c}}(P_{i,t}^{\mathrm{g}}) + G_i^{\mathrm{b}}(P_{i,t}^{\mathrm{b}})\Big)\\ &- \sum_{q\in\mathcal{Q}} U_q^{\mathrm{w}}(w_{i,q,t}) + \sum_{j\in\mathcal{J}}\big(p_{j,t}\check{V}_{j,t}(p_{j,t}) - U_j^{\mathrm{v}}(p_{j,t}; V_{j,t})\big)\\ &+ \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} G_{i,j}^{\mathrm{d}}\left(v_{i,j,t} + \sum_{q\in\mathcal{Q}}\tilde{w}_{i,j,q,t}\right).\end{aligned} \tag{3.14}$$

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

$$\Psi^* := \min_{\{\mathbf{x}_t, \mathbf{Y}_t,\ \forall t\}} \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right] \tag{3.15a}$$

$$\text{s.t.}\quad Y^{\rm b}_{i,t+1} = Y^{\rm b}_{i,t} + P^{\rm b}_{i,t},\ \forall i,t \tag{3.15b}$$

$$\underline{Y}^{\rm b}_i \le Y^{\rm b}_{i,t} \le \overline{Y}^{\rm b}_i,\ \forall i,t \tag{3.15c}$$

$$\underline{P}^{\rm b}_i \le P^{\rm b}_{i,t} \le \overline{P}^{\rm b}_i,\ \forall i,t \tag{3.15d}$$

$$0 \le P^{\rm g}_{i,t} \le \overline{P}^{\rm g}_i,\ \forall i,t \tag{3.15e}$$

$$0 \le d_{i,t} \le M_i D_i,\ \forall i,t \tag{3.15f}$$

$$\textstyle\sum_{i\in\mathcal{I}_j} v_{i,j,t} = V_{j,t} - p_{j,t}/(2\kappa_j),\ \forall j,t \tag{3.15g}$$

$$0 \le p_{j,t} \le 2\eta\kappa_j V_{j,t},\ \forall j,t \tag{3.15h}$$

$$v_{i,j,t} + \textstyle\sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t} \le B_{j,i},\ \forall j,i \tag{3.15i}$$

$$d_{i,t} = \textstyle\sum_{j\in\mathcal{J}} v_{i,j,t} + \sum_{q\in\mathcal{Q}} w_{i,q,t},\ \forall i,t \tag{3.15j}$$

$$Y^{\rm mn}_{j,q,t+1} = \left[Y^{\rm mn}_{j,q,t} + W_{j,q,t} - \textstyle\sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\right]_0^{\infty},\ \forall j,q,t \tag{3.15k}$$

$$Y^{\rm dc}_{i,q,t+1} = \left[Y^{\rm dc}_{i,q,t} - w_{i,q,t} + \textstyle\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t}\right]_0^{\infty},\ \forall i,q,t \tag{3.15l}$$

$$Y^{\rm mn}_{j,q,t} < \infty,\ \forall j,q,t;\ Y^{\rm dc}_{i,q,t} < \infty,\ \forall i,q,t; \tag{3.15m}$$

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 $\boldsymbol{s}_t$). Although strictly speaking the problem in (5.7) is convex, the battery dynamics in (4.21b) as well as the delay-tolerant workload queues in (5.7c) and (4.21k) couple the optimization variables over the infinite time horizon. Even worse, for the practical case where the knowledge of $\mathbf{s}_t$ is causal, finding the optimal solution requires using dynamic programming tools, which are generally intractable. Our approach to circumventing this obstacle is to relax (4.21b), (5.7c) and (4.21k), by replacing them with average constraints, and employ dual decomposition techniques to separate the solution across time. This is elaborated in the next section.

3.2.1 Problem relaxation

Combining (5.7c), (4.21k) and (3.15m), it follows that in the long term the average workload arrival and departure rates must satisfy the following necessary conditions

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[W_{j,q,t}\right]\leq\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q,t}\right],\ \forall j,q \tag{3.16a}$$

and

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}\right]\leq\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[w_{i,q,t}\right],\ \forall i,q. \tag{3.16b}$$

In words, in the long term all buffered DWs should be served. Upon observing that the batteries in (4.21b) and (3.15c) exhibit dynamics very similar to those of the workload queues, we use the same relaxation and require

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[P^{\mathrm{b}}_{i,t}\right]=0,\ \ \forall i. \tag{3.16c}$$

As before, (3.16c) guarantees that in the long term the energy stored into the battery and the energy taken from the battery are equal. Using (3.16a)-(3.16c), we can write the relaxed version of (5.7) as

$$\tilde{\Psi}^{*}:=\min_{\{\mathbf{x}_t\}}\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right] \tag{3.17}$$

$$\text{s.t.}\ \ (3.15d)-(4.21j),\ (3.16a)-(3.16c).$$

Compared to (5.7), variables $\mathbf{Y}_t:=\{Y^{\mathrm{b}}_{i,t},Y^{\mathrm{mn}}_{j,q,t},Y^{\mathrm{dc}}_{i,q,t},\forall i,j,q\}$ are not present in (5.9), and the time-coupling constraints (4.21b) and (5.7c)-(4.21k) are replaced with (3.16a)-(3.16c).

The problem in (5.9) has a number of interesting properties, including: a) since (5.9) is a relaxed version of (5.7), 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 Sec. 3.3.5, there exist low-complexity solvers that approximate the solution of (5.9) while being feasible for (5.7).

Regarding property b), using arguments similar to those in, e.g. [54, 72], it can be shown that if the random process $\boldsymbol{s}_t$ is stationary, there exists a stationary control policy $\boldsymbol{x}_t(\boldsymbol{s}_t)$ that: is a pure function of the current $\boldsymbol{s}_t$; satisfies (3.15d)-(4.21j); and guarantees that $\mathbb{E}[\Psi_t(\mathbf{x}_t(\mathbf{s}_t))]=\tilde{\Psi}^{*}$, $\mathbb{E}[W_{j,q,t}(\mathbf{s}_t)-\sum_{i\in\mathcal{I}_j}\tilde{w}_{i,j,q,t}(\mathbf{s}_t)]\leq 0,\ \forall j,q$, $\mathbb{E}[w_{i,q,t}(\mathbf{s}_t)-\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}(\mathbf{s}_t)]\leq 0,\ \forall i,q$ and

$\mathbb{E}[P^{\mathrm{b}}_{i,t}(\mathbf{s}_t)] = 0,\ \forall i$ [cf. (3.16a)-(3.16c)]. This implies that the limiting time averages in (5.9) 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 (3.16a)-(3.16c), we will dualize the long-term constraints (3.16a)-(3.16c), 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.

3.2.2 Dual decomposition

Let $\{\lambda^{\mathrm{mn}}_{j,q}\}$, $\{\lambda^{\mathrm{dc}}_{i,q}\}$ and $\{\lambda^{\mathrm{b}}_{i}\}$ denote the Lagrange multipliers associated with constraints (3.16a), (4.5) and (3.16c), respectively. With $\mathbf{x} := \{\mathbf{x}_t, \forall t\}$, and $\boldsymbol{\varpi}$ collecting all the multipliers, the partial Lagrangian function of (5.9) is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\varpi}) := \mathbb{E}[\Psi_t(\mathbf{x}_t)] + \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}} \mathbb{E}\left[\lambda^{\mathrm{dc}}_{i,q}\left(\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} - w_{i,q,t}\right)\right]$$
$$+ \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}} \mathbb{E}\left[\lambda^{\mathrm{mn}}_{j,q}\left(W_{j,q,t} - \sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\right)\right] + \sum_{i\in\mathcal{I}} \mathbb{E}\left[\lambda^{\mathrm{b}}_{i}\left(P^{\mathrm{b}}_{i,t}\right)\right]. \tag{3.18}$$

With $\mathcal{X}_t$ denoting feasible set defined by the instantaneous constraints (3.15d)-(4.21j), which are the ones not dualized in (5.11), the Lagrange dual function is

$$\mathcal{D}(\boldsymbol{\varpi}) := \min_{\{\mathbf{x}_t\in\mathcal{X}_t\}_{t\in\mathcal{T}}} \mathcal{L}(\mathbf{x}, \boldsymbol{\varpi}) \tag{3.19}$$

and the dual problem of (5.9) is

$$\max_{\boldsymbol{\varpi}} \mathcal{D}(\boldsymbol{\varpi}). \tag{3.20}$$

For the dual problem (5.13), a standard subgradient iteration can be employed to obtain the optimal $\boldsymbol{\varpi}^*$. Namely, with k denoting an iteration index, the multipliers at iteration $k+1$, denoted by $\boldsymbol{\varpi}(k+1)$, are found as

$$\lambda^{\mathrm{dc}}_{i,q}(k+1) = \left[\lambda^{\mathrm{dc}}_{i,q}(k) + \mu g_{\lambda^{\mathrm{dc}}_{i,q}}(k)\right]^{\infty}_{0},\ \forall i, q \tag{3.21a}$$

$$\lambda^{\mathrm{mn}}_{j,q}(k+1) = \left[\lambda^{\mathrm{mn}}_{j,q}(k) + \mu g_{\lambda^{\mathrm{mn}}_{j,q}}(k)\right]^{\infty}_{0},\ \forall j, q \tag{3.21b}$$

$$\lambda^{\mathrm{b}}_{i}(k+1) = \lambda^{\mathrm{b}}_{i}(k) + \mu g_{\lambda^{\mathrm{b}}_{i}}(k),\ \forall i \tag{3.21c}$$

where $\mu > 0$ is a constant stepsize that, if convenient, can be rendered different for each multiplier, and $g_{\boldsymbol{\varpi}}(k) := \{g_{\lambda_i^{\mathrm{b}}}(k), g_{\lambda_{i,q}^{\mathrm{dc}}}(k), g_{\lambda_{j,q}^{\mathrm{mn}}}(k),\ \forall i,j,q\}$ denote the subgradients of $\mathcal{D}(\boldsymbol{\varpi})$ in (5.12) w.r.t. the corresponding dual variables. These can be expressed as $g_{\lambda_{i,q}^{\mathrm{dc}}}(k) = \mathbb{E}\Big[\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t}(k) - w_{i,q,t}(k)\Big]$, $g_{\lambda_{j,q}^{\mathrm{mn}}}(k) = \mathbb{E}\Big[W_{j,q,t}(k) - \sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}(k)\Big]$, and, $g_{\lambda_i^{\mathrm{b}}}(k) = \mathbb{E}\Big[P_{i,t}^{\mathrm{b}}(k)\Big]$, with $\mathbf{x}(k)$ standing for the primal minimizers of the Lagrangian for the k-th iteration of the subgradient method, i.e., $\mathbf{x}(k) := \arg\min_{\mathbf{x}}\ \mathcal{L}(\mathbf{x}, \boldsymbol{\varpi}(k))$ subject to (3.15d)-(4.21j).

Due to the linearity of the expectation operator, the minimization w.r.t. the primal variables in (5.12) can be performed *separately across time*. Hence, the primal minimizers $\mathbf{x}(k) = \{\mathbf{x}_t(k)\}_{t\in\mathcal{T}}$ can be found by solving the following (infinitely many) instantaneous sub-problems, one per t

$$\begin{aligned}\mathbf{x}_t(k) \in \operatorname*{argmin}_{\mathbf{x}_t}\ \Psi_t(\mathbf{x}_t) &+ \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}} \lambda_{i,q}^{\mathrm{dc}}(k)\left(\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} - w_{i,q,t}\right) \\ &+ \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}} \lambda_{j,q}^{\mathrm{mn}}(k)\left(W_{j,q,t} - \sum_{i\in\mathcal{I}} \tilde{w}_{i,j,q,t}\right) + \sum_{i\in\mathcal{I}} \lambda_i^{\mathrm{b}}(k) P_{i,t}^{\mathrm{b}} \\ &\text{s.t. } (3.15d) - (4.21j)\end{aligned} \tag{3.22}$$

where the operator $\in$ accounts for cases that the Lagrangian has more than one minimizer. The minimization in (4.28) is convex and has a low dimensionality, so that 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.

3.2.3 Stochastic dual subgradient

The standard dual subgradient iterations (4.26) involve taking the expectation over the stationary distribution of $\mathbf{s}_t$ to obtain the subgradient $g_{\boldsymbol{\varpi}}(k)$. 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 [65]. The benefits are multiple, including: a) considerably reduced computational complexity; b) the distribution of $\mathbf{s}_t$ need not be known; and, c) the algorithms will be robust to noise and non-stationary environments.

Specifically, the iterations in (4.26) are replaced with

$$\lambda^{\mathrm{b}}_{i,t+1} = \lambda^{\mathrm{b}}_{i,t} + \mu P^{\mathrm{b}}_{i,t},\ \forall i \tag{3.23a}$$

$$\lambda^{\mathrm{dc}}_{i,q,t+1} = \left[\lambda^{\mathrm{dc}}_{i,q,t} + \mu\Big(\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} - w_{i,q,t}\Big)\right]^{\infty}_{0},\ \forall i,q \tag{3.23b}$$

$$\lambda^{\mathrm{mn}}_{j,q,t+1} = \left[\lambda^{\mathrm{mn}}_{j,q,t} + \mu\Big(W_{j,q,t} - \sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\Big)\right]^{\infty}_{0},\ \forall j,q \tag{3.23c}$$

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

$$\begin{aligned}\min_{\mathbf{x}_t}\ \Phi(\mathbf{x}_t) := \Psi_t(\mathbf{x}_t) + \sum_{i\in\mathcal{I}}\left[\sum_{q\in\mathcal{Q}} \lambda^{\mathrm{dc}}_{i,q,t}\left(\sum_{j\in\mathcal{J}} \tilde{w}_{i,j,q,t} - w_{i,q,t}\right) + \lambda^{\mathrm{b}}_{i,t} P^{\mathrm{b}}_{i,t}\right] \\ + \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}} \lambda^{\mathrm{mn}}_{j,q,t}\left(W_{j,q,t} - \sum_{i\in\mathcal{I}_j} \tilde{w}_{i,j,q,t}\right) \\ \text{s.t.}\quad (3.15d) - (4.21j).\end{aligned} \tag{3.24}$$

As will be shown in Section 3.3.5, the stochastic iterations in (3.23) and (3.24) provide two additional benefits critical for the problem at hand. First, there are performance and feasibility guarantees establishing that the solution provided by (3.24) is a tight approximation to the solution of (5.9). Second, if properly initialized, the solution provided by (3.24) can be shown to be feasible for the original problem in (5.7). Last but not least, links between the stochastic estimates in (3.23) and the battery and queue lengths can be established; see [29, 50] for a rigorous discussion.

Remark 2. In practice, it can be useful to rescale the subgradient in (3.23) 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 is very different, stepsize adjustment facilitates numerical convergence. Second, within one class of constraints – for example, flow conservation at the MN side – using different stepsizes offers as a mechanism to effect delay or queuing priorities [50].

3.3 Real-Time Distributed Load Balancing

Once the optimization has been separated across time instants, 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 robustness and privacy [14].

Toward these objectives, we will again rely on dual decomposition methods. Specifically, we further dualize the instantaneous constraint (4.21j) in (3.24), which couples the optimization variables among MNs and DCs. The fact that the constraint is instantaneous means that it has to be satisfied at each and every time instant. As a result, the algorithms developed in this section will have to run several iterations per time instant (those can be thought of as micro-slots), and the overall network optimization algorithm will operate in two timescales. With $\boldsymbol{\pi} := [\pi_1, \ldots, \pi_I]^\top$ denoting the instantaneous Lagrange multipliers associated with (4.21j) in (3.24), the partial Lagrangian of the instantaneous problem in (3.24) can be written as[1] [cf. $\Phi(\mathbf{x})$ in (3.24)]

$$\tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\pi}) := \Phi(\mathbf{x}) + \sum_{i \in \mathcal{I}} \pi_i \left(\sum_{j \in \mathcal{J}} v_{i,j} + \sum_{q \in \mathcal{Q}} w_{i,q} - d_i \right). \tag{3.25}$$

Hence, with $\tilde{\mathcal{X}}$ denoting feasible set defined by constraints (3.15d)-(5.7b), the Lagrange dual function of (3.24) is $\tilde{\mathcal{D}}(\boldsymbol{\pi}) := \min_{\mathbf{x} \in \tilde{\mathcal{X}}} \ \tilde{\mathcal{L}}(\mathbf{x}, \boldsymbol{\pi})$, and the dual problem is

$$\max_{\boldsymbol{\pi}} \ \tilde{\mathcal{D}}(\boldsymbol{\pi}). \tag{3.26}$$

It is worth stressing that different from the dual formulation in (5.13), which facilitates the implementation of stochastic approximation schemes, the goal of the dual relaxation in (5.22) 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 (5.22): a dual subgradient method that can be used for any convex formulation, and a dual *accelerated* gradient method that requires some additional assumptions.

3.3.1 Subgradient iteration

With ℓ denoting the iteration (micro-slot) index, the optimal $\boldsymbol{\pi}^*$ is found upon running

$$\pi_i(\ell+1) = \pi_i(\ell) + \mu(\ell) \left(\sum_{j \in \mathcal{J}} v_{i,j}(\ell) + \sum_{q \in \mathcal{Q}} w_{i,q}(\ell) - d_i(\ell) \right). \tag{3.27}$$

[1]For notational brevity, time index t is dropped throughout this section.

As before, $\mathbf{x}(\ell)$ stands for the minimizer of the Lagrangian when $\boldsymbol{\pi} = \boldsymbol{\pi}(\ell)$. Specifically, at each iteration ℓ, every DC needs to obtain a tentative power allocation $\{d_i(\ell), P_i^{\mathrm{g}}(\ell), P_i^{\mathrm{b}}(\ell)\}$ by solving

$$\min_{d_i, P_i^{\mathrm{g}}, P_i^{\mathrm{b}}} \; G_i^{\mathrm{e}}(P^i) + G_i^{\mathrm{c}}(P_i^{\mathrm{g}}) + G_i^{\mathrm{b}}(P_i^{\mathrm{b}}) + \lambda_i^{\mathrm{b}} P_i^{\mathrm{b}} - \pi_i(\ell) d_i$$
$$\text{s.t. } (3.15d) - (4.21e) \tag{3.28}$$

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

$$\min_{w_{i,q}} \; \left(\pi_i(\ell) - \lambda_{i,q}^{\mathrm{dc}}\right) w_{i,q} - U_q^{\mathrm{w}}(w_{i,q}), \tag{3.29}$$

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

$$\min_{p_j, v_{i,j}, \tilde{w}_{i,j,q}} \sum_{i \in \mathcal{I}_j} \left[\sum_{q \in \mathcal{Q}} \left(\lambda_{i,q}^{\mathrm{dc}} - \lambda_{j,q}^{\mathrm{mn}}\right) \tilde{w}_{i,j,q} + \pi_i(\ell) v_{i,j} + G_{i,j}^{\mathrm{d}} \left(v_{i,j} + \sum_{q \in \mathcal{Q}} \tilde{w}_{i,j,q} \right) \right] + \frac{(p_j)^2}{2\kappa_j} - U_j^{\mathrm{v}}(p_j; V_j)$$
$$\text{s.t. } \; (4.21c) - (5.7b). \tag{3.30}$$

The dual subgradient method enjoys convergence guarantees if a sequence of non-summable diminishing stepsizes is chosen to satisfy $\lim_{\ell \to \infty} \mu(\ell) = 0$ and $\sum_{\ell=0}^{\infty} \mu(\ell) = \infty$ [15]. Since (3.24) is convex, the duality gap is zero, and the minimizer of the Lagrangian yields the optimal solution to the primal problem (3.24). Alternatively, if a constant stepsize μ is adopted, the subgradient iterations (3.27) are guaranteed to converge to a neighborhood of the optimal $\boldsymbol{\pi}^*$ for the dual problem (5.22), and the running average of the primal variables will converge to the optimal solution [15]. In practice, the iterations can be stopped once a pre-specified tolerance (or duality gap) is met. It is also worth noting that the dual subgradient update 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].

3.3.2 Fast Iterative Shrinkage-Thresholding Algorithm

Although the subgradient iteration is widely employed in a variety of applications, it does not fully leverage particular properties that the problem at hand may have, including the differentiability of the dual function and the Lipschitz continuity of its gradient. In this subsection, we develop an alternative approach for solving the dual problem based on the *Fast Iterative Shrinkage-Thresholding*

Algorithm 2 Dual FISTA Iteration for (5.22)

1: **Initialize:** with a proper $\boldsymbol{\pi}(0)$ and stepsize β
2: **for** $\ell = 1, 2 \dots$ **do**
3: MNs acquire $\boldsymbol{\gamma}(\ell)$
4: Solve (3.28)-(3.30) to obtain the virtual decision $\mathbf{x}(\boldsymbol{\gamma}(\ell))$
5: Update Lagrange multipliers $\boldsymbol{\pi}(\ell)$ via (3.31a)
6: Update $\boldsymbol{\theta}(\ell+1)$ and $\boldsymbol{\gamma}(\ell+1)$ via (3.31b)-(3.31c)
7: DCs send $\boldsymbol{\gamma}(\ell+1)$ to MNs
8: **end for**

Algorithm, which enjoys convergence rate faster than that of the traditional (sub-)gradient iterations [12].

Per iteration (micro-slot) ℓ, FISTA implements the following updates

$$\pi_i(\ell) = \gamma_i(\ell) + \beta\left(\sum_{j\in\mathcal{J}} v_{i,j}(\ell) + \sum_{q\in\mathcal{Q}} w_{i,q}(\ell) - d_i(\ell)\right) \tag{3.31a}$$

$$\theta_i(\ell+1) = \frac{1+\sqrt{1+4(\theta_i(\ell))^2}}{2} \tag{3.31b}$$

$$\gamma_i(\ell+1) = \pi_i(\ell) + \frac{\theta_i(\ell)-1}{\theta_i(\ell+1)}\left(\pi_i(\ell) - \pi_i(\ell-1)\right). \tag{3.31c}$$

Each DC will obtain $\{w_{i,q}(\ell), d_i(\ell), P_i^{\mathrm{g}}(\ell), P_i^{\mathrm{b}}(\ell)\}$ by solving (3.28) and (3.29) with $\boldsymbol{\pi}(\ell)$ replaced by $\boldsymbol{\gamma}(\ell)$, while each MN will obtain $\{v_{i,j}(\ell), \tilde{w}_{i,j,q}(\ell)\}$ by solving (3.30) with $\boldsymbol{\pi}(\ell)$ replaced by $\boldsymbol{\gamma}(\ell)$.

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)*** *the workload distribution cost functions $G_{i,j}^{\mathrm{d}}(\,\cdot\,)$ are strongly convex.* Note that from an engineering perspective, this strong convexity is reasonable. Oftentimes in practice the marginal rewards (costs) are *monotonically* decreasing (increasing), which guarantees strict convexity. But even if they are not, one can approximate $G_{i,j}^{\mathrm{d}}(v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t})$ by a strongly convex function $\tilde{G}_{i,j}^{\mathrm{d}}(v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t}) := G_{i,j}^{\mathrm{d}}(v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t}) + \epsilon\|v_{i,j,t}\|^2 + \epsilon\sum_{q\in\mathcal{Q}}\|\tilde{w}_{i,j,q,t}\|^2$, with $\epsilon > 0$. Differentiability of the

dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ follows from *AS1)*, as formally stated next.

Proposition 1. *For a given $\boldsymbol{\pi}$, the partial Lagrangian (3.25) has a unique minimizer; thus, the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ is continuously differentiable.*

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 the differentiability of a convex function and the strong convexity of its conjugate to facilitate the derivation of the Lipschitz constant; see [13, Lemma II.1].

Lemma 6. *Let h: $\mathbb{R}^n \to (-\infty, \infty]$ be a proper, lower semicontinuous, convex function, and let constant $\sigma > 0$. The following statements are equivalent: **S1)** Function h is differentiable and its gradient mapping ∇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 6, 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*

$$L := (J+Q+1) \times \max_{q,j,i} \left\{ \frac{1}{\sigma(U_q^{\mathrm{w}})}, \frac{1}{\sigma(G_i^{\mathrm{b}})}, \frac{1}{\sigma(G_{i,j}^{\mathrm{d}})}, \frac{1}{\sigma(G_i^{\mathrm{c}})}, \frac{M_i D_i^2}{2\underline{\alpha}_i^{\mathrm{s}} \varrho}, \frac{2\kappa_j^2}{2\kappa_j + \sigma(U_j^{\mathrm{v}})} \right\} \tag{3.32}$$

where $\sigma(U_q^{\mathrm{w}})$, $\sigma(G_i^{\mathrm{b}})$, $\sigma(G_{i,j}^{\mathrm{d}})$, $\sigma(G_i^{\mathrm{c}})$, $\sigma(U_j^{\mathrm{v}})$ are defined in Proposition 6 in Appendix A.

Proof. See Appendix A. □

With the definition of L, we are ready to establish the convergence result [13, 55], which closes this section.

Proposition 3. *If L denotes the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ in (3.32), and the step-size $\beta \in (0, \frac{1}{L}]$, then Algorithm 7 converges to optimal dual variable $\boldsymbol{\pi}^*$. And for $\ell \geq 1$, it satisfies*

$$\tilde{\mathcal{D}}(\boldsymbol{\pi}^*) - \tilde{\mathcal{D}}(\boldsymbol{\pi}(\ell)) \leq \frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2}{\beta(\ell+1)^2}. \tag{3.33}$$

3.3.3 Diagonal weighted FISTA

Computing the Lipschitz constant L, whose value has to be known to set β in (3.31), requires in general communication among all the DCs and MNs [cf. (3.32)], 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 $\mathbf{\Lambda}$ whose ith diagonal element is given by

$$\Lambda_{ii} = \sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}_j} \left\{ \frac{1}{\sigma(G^{\mathrm{d}}_{i,j})}, \frac{2\kappa_j^2}{2\kappa_j+\sigma(U^{\mathrm{v}}_j)} \right\} + \sum_{q\in\mathcal{Q}} \frac{1}{\sigma(U^{\mathrm{w}}_q)} + \max\left\{ \frac{1}{\sigma(G^{\mathrm{b}}_i)}, \frac{1}{\sigma(G^{\mathrm{c}}_i)}, \frac{M_i D_i^2}{2\underline{\alpha}^{\mathrm{s}}_i \varrho} \right\}. \tag{3.34}$$

Compared to the standard FISTA in Section 3.3.2, here each DC only needs to know the global information $\sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}_j} \left\{1/\sigma(G^{\mathrm{d}}_{i,j}), 2\kappa_j^2/\Big(2\kappa_j+\sigma(U^{\mathrm{v}}_j)\Big)\right\}$, which can be obtained from the subset of MNs the particular DC is connected to.

With the diagonal scaling matrix $\mathbf{\Lambda}$ defined in (3.34), we can consequently establish the next proposition.

Proposition 4. *If the update for $\pi_i(\ell)$ in (3.31a) is replaced with*

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

then Algorithm 7 converges to the optimal dual variable $\boldsymbol{\pi}^*$. And for $\ell \geq 1$, it holds that

$$\tilde{\mathcal{D}}(\boldsymbol{\pi}^*) - \tilde{\mathcal{D}}(\boldsymbol{\pi}(\ell)) \leq \frac{2\|\boldsymbol{\pi}^* - \boldsymbol{\pi}(0)\|^2_{\mathbf{\Lambda}}}{(\ell+1)^2}. \tag{3.35}$$

Proof. See Appendix B. □

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

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

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

3.3.4 Real-time distributed implementation

In this subsection, we are ready to introduce our *Distributed Geographical Load Balancing* algorithm (**DGLB**), which integrates the stochastic dual subgradient approach in Sec. 3.2.3, with the dual FISTA algorithm in Sec. 3.3.3.

It is also worth mentioning that when the real-time energy purchase and selling prices are identical at each DC [27], i.e., $\alpha^{\mathrm{p}}_{i,t} = \alpha^{\mathrm{s}}_{i,t}$, the DC subproblems (3.28)–(3.29) can be solved in closed-form, as formalized next.

Proposition 5. *The minimizers of DC subproblems (3.28) and (3.29) are*

$$P^{\mathrm{b}}_i(\ell) = \left[(\nabla G^{\mathrm{b}}_i)^{-1}(-\lambda^{\mathrm{b}}_i - \alpha^{\mathrm{p}}_{i,t})\right]^{\overline{P}^{\mathrm{b}}_i}_{\underline{P}^{\mathrm{b}}_i},\ P^{\mathrm{g}}_i(\ell) = \left[(\nabla G^{\mathrm{g}}_i)^{-1}(\alpha^{\mathrm{p}}_{i,t})\right]^{\overline{P}^{\mathrm{g}}_i}_{0},$$

and

$$d_i(\ell) = \left[\frac{M_i D_i^2 \pi_i(\ell)}{2\varrho(1+e_{i,t})\alpha^{\mathrm{p}}_{i,t}}\right]^{M_i D_i}_{0},\ w_{i,q,t}(\ell) = \left[(\nabla U^{\mathrm{w}}_q)^{-1}(\pi_i(\ell) - \lambda^{\mathrm{dc}}_{i,q})\right]^{\infty}_{0}.$$

Notice that the communication overhead of DGLB is fairly low. While DC sub-problems have closed-form solutions, MN sub-problems (3.30) can be solved in parallel, reducing the per-iteration complexity. In addition, leveraging the accelerated method (FISTA), the algorithm will usually converge within tens of iterations.

3.3.5 Performance guarantees

To arrive at our main claim, we begin by quantifying the optimality gap of the proposed DGLB. Based on the results [50, 72], the following lemma holds true.

Lemma 7. *If the random state $\boldsymbol{s}_t$ is either i.i.d. or follows a finite state ergodic Markov chain*[2]*, 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*

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

[2]Here the ergodicity means that the stochastic process $\{\boldsymbol{s}_t\}$ is stationary, positive recurrent, and irreducible.

Algorithm 3 Distributed Geographical Load Balancing

1: **Initialize** Lagrange multipliers $\boldsymbol{\varpi}_0$, and stepsizes $\mu, \boldsymbol{\Lambda}$.

2: **Per slot** t, observe $\boldsymbol{\varpi}_t, \mathbf{s}_t$, with $\mathbf{s}_t$ collecting all random variables at time t, and then run the following tasks.

3: **Communication overhead**. Obtain $\boldsymbol{\pi}_t^*$ by using the subgradient update (3.27) or FISTA updates (3.31).

4: **MN pricing and routing**. Each MN solves (3.30) 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 (3.29) 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 $\{d_{i,t}, P_{i,t}^{\mathrm{g}}, P_{i,t}^{\mathrm{b}}\}$ by solving (3.28). Perform energy transaction with the main grid; that is, buy the energy amount $[P_{i,t}^{\mathrm{m}}]^+$ with price $\alpha_{b,t}^t$ upon energy deficit, or, sell the energy amount $[P_{i,t}^{\mathrm{m}}]^-$ with price $\alpha_{s,t}^t$ upon energy surplus. Perform battery (dis)charging according to $P_{i,t}^{\mathrm{b}}$, and plan CG generations.

7: **Lagrange multiplier updates**. With $\{P_{i,t}^{\mathrm{b}}, \tilde{w}_{i,j,q,t}, w_{i,q,t}\}$ available, DCs update Lagrange multipliers $\{\lambda_{i,t+1}^{\mathrm{b}}\}$ and $\{\lambda_{i,q,t+1}^{\mathrm{dc}}\}$ via (3.23a)-(3.23b), and MNs update Lagrange multipliers $\{\lambda_{j,q,t+1}^{\mathrm{mn}}\}$ via (3.23c).

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

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

and Ψ^* is the optimal value of (5.7) under any feasible control.

Proof. See Appendix C. □

Lemma 7 asserts that the proposed scheme can achieve a near-optimal objective value for (5.7). However, since the proposed algorithm approximates a relaxation of (5.7) [cf. (5.9)], the resultant dynamic control policy is not guaranteed to be feasible. In the sequel, we will establish that, if

properly initialized, DGLB indeed yields a feasible policy for (5.7). To achieve this, we start by characterizing a pair of properties of the optimal policy.

Lemma 8. *If* $\overline{\alpha}_i^{\mathrm{p}} := \max\{\alpha_{i,t}^{\mathrm{p}}, \forall t\}$ *and* $\underline{\alpha}_i^{\mathrm{s}} := \min\{\alpha_{i,t}^{\mathrm{s}}, \forall t\}$*, the real-time battery (dis)charging decisions* $P_{i,t}^{\mathrm{b}}$ *generated by the DGBL algorithm satisfy: i)* $P_{i,t}^{\mathrm{b}}(\boldsymbol{\varpi}_t) = \underline{P}_i^{\mathrm{b}}$*, if* $\lambda_{i,t}^{\mathrm{b}} > -\underline{\alpha}_i^{\mathrm{s}} - \partial \underline{G}_i^{\mathrm{b}}$*; and, ii)* $P_{i,t}^{\mathrm{b}}(\boldsymbol{\varpi}_t) = \overline{P}_i^{\mathrm{b}}$*, if* $\lambda_{i,t}^{\mathrm{b}} < -\overline{\alpha}_i^{\mathrm{p}} - \partial \overline{G}_i^{\mathrm{b}}$.

Lemma 8 characterizes the battery (dis-)charging behavior, which allows us to establish the next result.

Lemma 9. *If the stepsize satisfies* $\mu \geq \underline{\mu}$*, where*

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

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

These two lemmas are generalizations of [21, Lemma 4] and [21, Lemma 5]; their proof is omitted here for brevity. Consider now the linear mapping

$$Y_{i,t}^{\mathrm{b}} = \frac{\lambda_{i,t}^{\mathrm{b}}}{\mu} + \frac{\overline{\alpha}_i^{\mathrm{p}}}{\mu} + \frac{\partial \overline{G}_i^{\mathrm{b}}}{\mu} + \underline{Y}_i^{\mathrm{b}} - \underline{P}_i^{\mathrm{b}}, \quad \forall i. \tag{3.36}$$

It can be readily seen from Lemma 9 that $\underline{Y}_i^{\mathrm{b}} \leq Y_{i,t}^{\mathrm{b}} \leq \overline{Y}_i^{\mathrm{b}}$ holds for all i and t; i.e., (3.15c) are always satisfied under the proposed online scheme. With the battery (dis)charging dynamics (4.21b) naturally performed, feasibility of the control actions $\mathbf{x}(\boldsymbol{\varpi}_t)$ can be maintained for the original problem, provided that we select a stepsize $\mu \geq \underline{\mu}$.

Based on Lemmas 7 and 9, we are able to reach the following main result.

Theorem 2. *Upon setting* $\lambda_{i,0}^{\mathrm{b}} = \mu Y_{i,0}^{\mathrm{b}} - \mu \underline{Y}_i^{\mathrm{b}} - \overline{\alpha}_i^{\mathrm{p}} - \partial \overline{G}_i^{\mathrm{b}} + \mu \underline{P}_i^{\mathrm{b}}$, $\forall i$*, and selecting a stepsize* $\mu \geq \underline{\mu}$*, the DGLB algorithm yields a feasible dynamic control scheme for (5.9), which satisfies*

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

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

The theorem states that with a proper initialization, the proposed online policy using any $\mu \geq \underline{\mu}$ will be feasible for (5.7) and with bounded optimality loss. Choosing $\mu = \underline{\mu}$, the minimum optimality gap between the online DGLB and the offline scheduling is given by $\underline{\mu} M \mathbb{E}[\Delta T_n^2]/\mathbb{E}[\Delta T_n]$. Scenarios where both the difference between purchase and selling prices $(\overline{\alpha}_i^{\mathrm{p}} - \underline{\alpha}_i^{\mathrm{s}})$ and the difference between marginal charging and discharging costs $\partial \overline{G}_i^{\mathrm{b}} - \partial \underline{G}_i^{\mathrm{b}}$ approach zero will allow for selecting $\underline{\mu}$ very small [cf. Lemma 9], so that the optimality loss is practically zero. The same is true in scenarios where the battery capacities $\overline{Y}_i^{\mathrm{b}}$ are very large. This makes sense intuitively because as both $(\overline{\alpha}_i^{\mathrm{p}} - \underline{\alpha}_i^{\mathrm{s}})$ and $\partial \overline{G}_i^{\mathrm{b}} - \partial \underline{G}_i^{\mathrm{b}}$ approach zero, purchasing extra energy to charge the batteries (if they are close to empty) or selling it to discharge them (if they are close to full) will always be profitable. Similarly, when batteries have large capacity, the upper bounds in (3.15c) do not hold as equalities, and stationarity policies obeying the long-term energy conservation constraint will be optimal. 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 [21, 50].

Remark 3. 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 [11, 31]. Numerical results will further demonstrate that the DGLB can obtain a desirable performance even in real data scenarios.

Remark 4. Readers familiar with the so-called Lyapunov-optimization (LO) framework can recognize similarities between the stochastic dual subgradient (SDGD) solver proposed here, and the tools in [54, 72]. The differences between the two methods can be summarized as follows:

D1) The LO solver relies on the so-called "virtual queues" to ensure that long-term average constraints are met, where the tuning parameter V in [54, 72] corresponds to the inverse of the stepsize μ in the SDGD setup. In contrast, "virtual queues" emerge naturally as Lagrange multiplier iterations in our SDGD algorithms;

D2) Leveraging duality and stochastic approximation techniques, the SDGD iteration is also easy to interpret. The multipliers for instance, can be viewed as the instantaneous charging prices, which reveals the intuition behind workload routing, scheduling and real-time (dis)charging deci-

sions, 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 SDGD algorithms [cf. Lemma 3].

3.4 Numerical Tests

This section presents numerical test cases to confirm the analytical claims in Section 3.3.5, and demonstrate the merits of the proposed approach. We start by describing the simulation setup. The Matlab package `CVX 2.1` [5] is used to solve the optimization problems. The duration of a scheduling period (time slot) is one hour. 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 $\{M_i\} = \{1000, 750, 750, 1000\}$. One unit of workload is assumed to require the computing resources of 5 servers. The value of D_i is set to 0.2 for all DCs 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 with support $[20, 300]$, and the communication cost function is $G^{\mathrm{d}}_{i,j}(\{\tilde{w}_{i,j,q,t}\}_{q\in\mathcal{Q}}, v_{i,j,t}) = c^{\mathrm{d}}_{i,j}(v_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}_{i,j,q,t})^2 + v^2_{i,j,t} + \sum_{q\in\mathcal{Q}} \tilde{w}^2_{i,j,q,t}$, with $c^{\mathrm{d}}_{i,j}$ inversely proportional to $B_{j,i}$. We consider $Q = 2$ types of DW jobs, with revenue function $U^{\mathrm{w}}_q(w_{i,q,t}) = -u_q(w_{i,q,t})^2 + 50u_q w_{i,q,t}$ and u_q being uniformly distributed within $[1, 3]$ cents/(unit)^2. The coefficient κ_j in $G^{\mathrm{u}}_j(\check{V}_{j,t})$ is generated from a uniform distribution within $[1, 3]$ cents/(unit)^2 too. 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 4.1. The CG cost is $G^{\mathrm{c}}_i(P^{\mathrm{g}}_{i,t}) = c^{\mathrm{g}}(P^{\mathrm{g}}_{i,t})^2 + 10c^{\mathrm{g}}P^{\mathrm{g}}_{i,t}$ with $c^{\mathrm{g}} = 0.5$ cents/(kWh)^2, while the battery (dis)charging cost is $G^{\mathrm{b}}_i(P^{\mathrm{b}}_{i,t}) = c^{\mathrm{b}}(P^{\mathrm{b}}_{i,t})^2$ with $c^{\mathrm{b}} = 1$ cents/(kWh)^2.

Two sets of numerical results are presented: one to demonstrate convergence and robustness in a synthetic scenario using i.i.d. random variables (Section 3.4.1), and the other one to illustrate performance in a practical scenario using real-world date (Section 3.4.2). Note that workload arrivals, energy prices, and renewable generations in Test Case 2 are highly correlated over time, so that it will serve to assess the applicability of DGLB to non-stationary setups. To benchmark performance

Table 3.1: DC power-related parameters. The units are kW or kWh.

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

Table 3.2: Averages of the time series used to run Test Case 2.

Index for DC i or MN j	1	2	3	4
Mean ($\alpha_{i,t}^{\mathrm{p}}$), cent/kWh	9.77	6.47	8.32	12.15
Mean ($P_{i,t}^{\mathrm{r}}$), 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

of the proposed algorithm, three baseline schemes are tested including both the local load balance (LLB) as well as the geographical load balancing (GLB) schemes.

1) **ALG 1** (LLB, with incentive payment, DW scheduling, RES and storages): ALG 1 is similar to the algorithms in [21, 72], 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 [82], 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.

3) **ALG 3** (GLB, without RES and storages, with DW scheduling and incentive payment): ALG 3 mimics the algorithm in [85], but with only one-timescale operation, where DCs are only powered by CG or power from the spot market, without considering RES and storage units.

3.4.1 Test Case 1: convergence and robustness

In the first test case, the purchase price $\alpha_{i,t}^{\mathrm{p}}$ is uniformly distributed within $[10, 30]$ \$/kWh, samples of the renewable supply $\{P_{i,t}^{\mathrm{r}}\}$ 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 j according to a Poisson process, all with average arrival rates 50 units/slot.

The convergence results when solving the real-time problems in (3.24) are compared in Fig. 3.2, where $T = 100$ slots are considered, each consisting of up to a 200 micro-slots. In the legend,

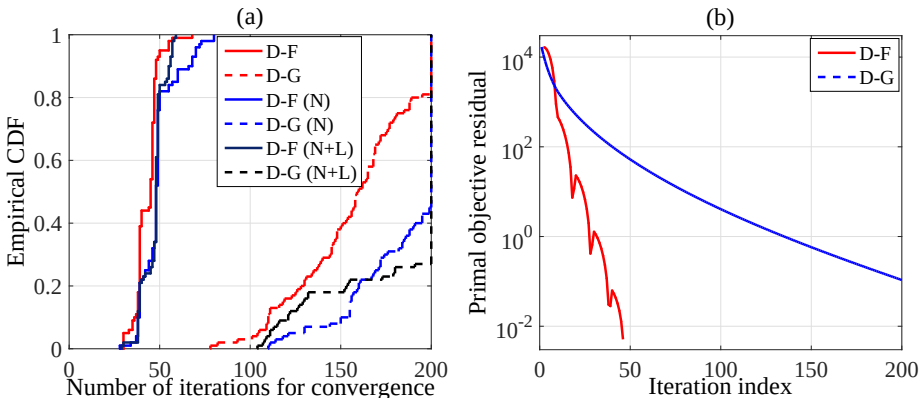

Figure 3.2: The left panel shows the empirical CDF of the number of iterations needed to converge, and the right panel plots the evolution of the primal objective residual for the noise-free case for one realization.

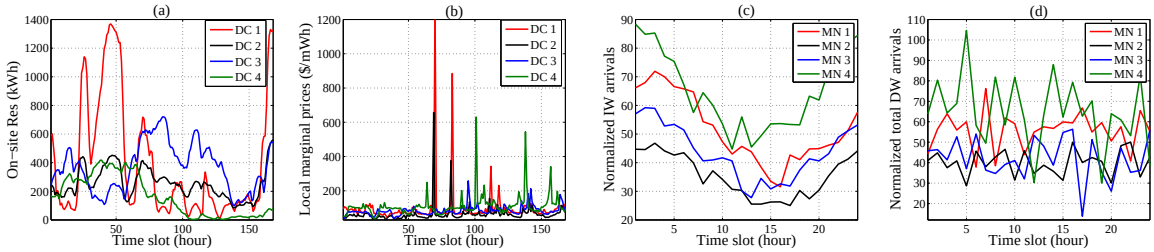

Figure 3.3: Time variation of the RES generation, local marginal prices, and IW and DW arrivals used in Test Case 2 [2, 4, 7, 22, 82].

"D" denotes distributed, "F" FISTA, "G" gradient, "N" the presence of *noise*, and "L" that some of the multipliers are *lost*. A stepsize $\mu(\ell) = 0.5/\sqrt{\ell}$, $\ell = 1, ..., 200$ is employed for the subgradient iteration [20], the diagonal-scaled stepsize (3.34) 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. 3.2(a)), 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 the dual FISTA is illustrated in Fig. 3.2(b), which depicts the evolution of the residual for both D-G and D-F for one of the 100 slots. The curves show that the residual reduction per update (micro-slot) is considerably larger for D-F. To demonstrate robustness of the distributed algorithms, the blue and black lines in Fig. 3.2(a) represent the empirical CDFs of iteration complexity when the exchanged multipliers are noisy or sporadically lost. Adding zero-mean Gaussian

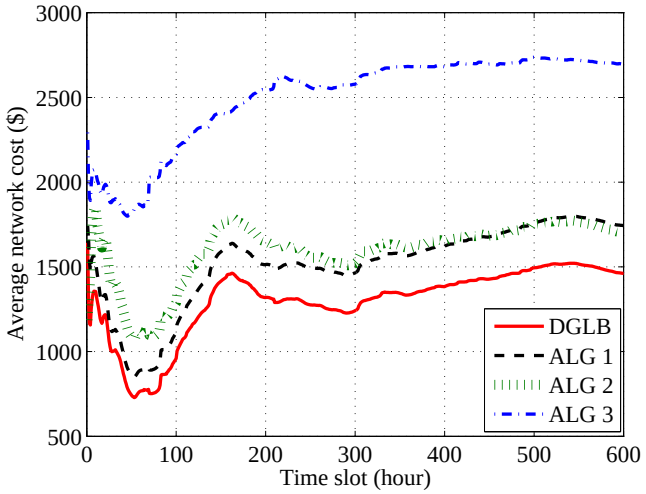

Figure 3.4: Comparison of time-average network costs in the DC network.

noise with variance $\sigma^2 = 1$, the blue lines show the performance of the two algorithms. The convergence results for a link outage probability 0.2 are further compared using the black lines[3]. The main observation is that the number of iterations required for the dual subgradient to converge ranges from 70 to more than 200, while in most cases the dual FISTA converges within as few as 45-55 iterations.

3.4.2 Test Case 2: scenario with real data

In this test case, the purchase prices per slot t are set equal to the selling prices; i.e., $\alpha^{\mathrm{p}}_{i,t} = \alpha^{\mathrm{s}}_{i,t}$, $\forall i, t$. The purchase prices at DC $1, 3-4$ are re-scaled from the local real-time hourly 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 [2, 4, 7]. 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, the IWs are generated by duplicating the Wikipedia trace over a 24-hour period [82], while the DWs are generated by copying the hourly MapReduce trace over a day [22]. In both cases, white Gaussian noise with variance randomly drawn between $3 - 5\mathrm{dB}$ was added to the original

[3]In the case of a link outage, the subproblems (3.28)-(3.30) at iteration ℓ are solved using the outdated Lagrange multipliers $\boldsymbol{\pi}_t(\ell - 1)$.

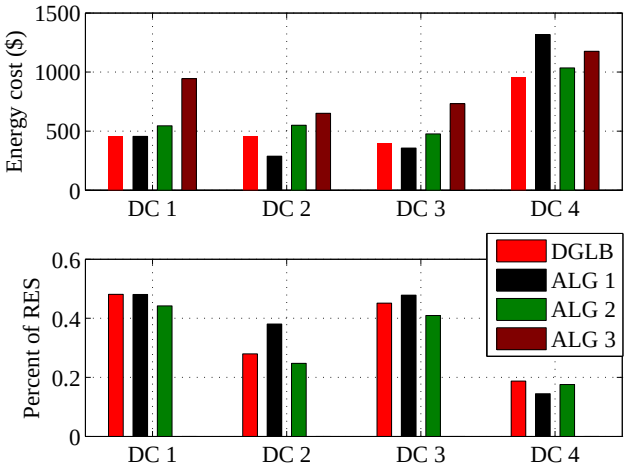

Figure 3.5: Comparison of energy cost and RES usage at each DC. The RES usage is the ratio of consumed RES to the total energy consumption.

values, which were also rescaled to model regional differences. The Western Time Zone (UTC-8) was used for time-keeping, and all real 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. 3.3(a)-(b), those of IWs and DWs over a day are shown in Figs. 3.3(c)-(d), and their average is listed in Table 3.2.

Fig. 3.4 depicts the evolution of the total network cost (primal objective) of DGLB and ALGs 1-4. Over $T = 600$ time slots, the average network cost of the DGLB algorithm is 17% lower than

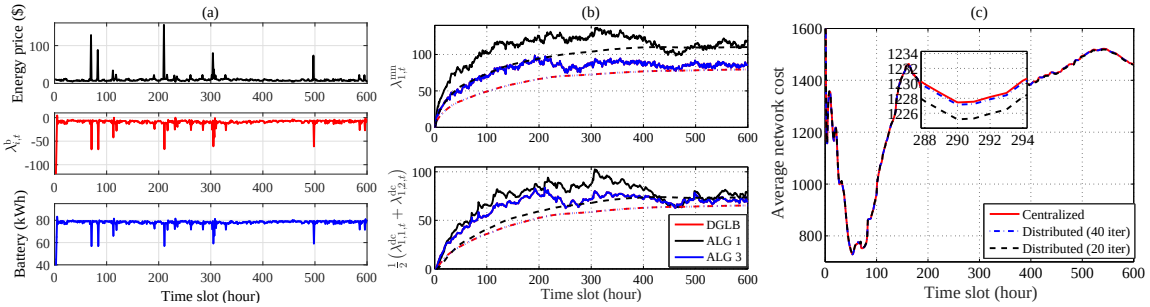

Figure 3.6: The left panel shows the evolutions of the price $\alpha^{\mathrm{p}}_{1,t}$, the battery level $C_{1,t}$, and the Lagrange multiplier $\lambda^{\mathrm{b}}_{1,t}$. The middle panel plots the evolutions of Lagrange multipliers $\lambda^{\mathrm{mn}}_{1,t}$ and $\left(\lambda^{\mathrm{dc}}_{1,1,t} + \lambda^{\mathrm{dc}}_{1,2,t}\right)/2$. The right panel compares the evolutions of network costs for DGLB using the distributed diagonal-weighted FISTA running 20 and 40 iterations.

that of ALGs 1-2, and around 46% lower than that of ALG 3. Recall that ALGs 1-2 are vulnerable to the high fluctuation of energy 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 as neither RES nor storage units are integrated to hedge against future high prices. Indeed, as corroborated by Fig. 3.4, ALGs 1-3 have to buy more (expensive) energy from the spot market to cope with the peaks of the demand. By contrast, the proposed DGLB algorithm takes advantage of the incentive payments, workload queues as well as RES and storage devices, so it can smooth the workload curvatures and use RES and stored energy to avoid future purchases at high prices, resulting in a smaller average network cost.

The average energy cost and the ratio of RES to the total energy consumption are plotted in Fig. 3.5. While the proposed DGLB generally incurs lower energy cost, ALG 2 (LLB) has the smallest energy cost and the largest percent of RES in DCs 2-3. This makes sense intuitively, because the workload demands in DCs 2-3 are relatively low [cf. Figs. 3.3 (c)-(d)]. While GLB policies smooth the load profile by allocating remote loads to MNs 2-3, LLB can only leverage local resources, leading to a high cost and low RES utilization at DC 4.

To better understand the role of the Lagrange multipliers in the workload and power balancing, the trajectories of some of them are depicted in Figs. 3.6(a)-(b), where the dashed lines in Fig. 3.6(b) are the running average of instantaneous multipliers. Specifically, to streamline the analysis, only multipliers associated with MN 1 and DC 1 are shown. The (negative) Lagrange multiplier $-\lambda^{\mathrm{b}}_{1,t}$ in the central panel of Fig. 3.6(a) is of particular interest, since it 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 (3.36); 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 in Fig. 3.6(b), which are associated with workload queues, can be related to the workload delay, or, the congestion price at each MN or DC. As verified in Fig. 3.6(b), the multipliers for DGLB and ALG 3, both of which use a GLB strategy, follow almost identical trajectories. Differently, ALG 1 exhibits larger delay, especially at the MN side. Intuitively, this is because in ALG 1 the 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. 3.6(b), ALG 2 indeed experiences a close-to-zero delay, at the expense of high network cost [cf. Fig. 3.4].

Finally, since in practice the DGLB algorithm may not be able to know the optimal value of (3.24) a-priori, we assess the performance of our algorithms when running a *fixed* number of iterations. Fig. 3.6(c) plots the average network costs of the centralized solver for (3.24) using `CVX` [5], as well as the DGLB algorithm when running 40 and 20 FISTA-iterations. 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). Combined with the results in Fig. 3.2, this experiment further demonstrates the merits of DGLB, and its suitability for distributed real-time implementation.

3.5 Appendices

3.5.1 Proof of Proposition 2

We first show the conjugate relationship between a σ-strongly convex function and the dual function $\tilde{\mathcal{D}}$, and then derive the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$ using the equivalence between the differentiability of $\tilde{\mathcal{D}}$ and the strong convexity of its conjugate. To that end, let us write the extended-valued function of the objective $\Phi(\mathbf{x})$ in (3.24) as $\tilde{\Phi}(\mathbf{x}) := \Phi(\mathbf{x})$ if $\mathbf{x} \in \tilde{\mathcal{X}}$, and $\tilde{\Phi}(\mathbf{x}) := \infty$ otherwise.

Using (3.11), variable $P^{\mathrm{m}}_{i,t}$ in $\mathbf{x}_t$ can be written as a function of other variables in $\mathbf{x}_t$ and $\mathbf{s}_t$. After doing this, each of the remaining terms in $\Phi(\mathbf{x})$ is *strongly* convex, and $\tilde{\Phi}(\mathbf{x})$ is strongly convex too. The constant quantifying the strong convexity of $\tilde{\Phi}(\mathbf{x})$ is provided next.

Proposition 6. *The extended-valued function $\tilde{\Phi}$ is $\sigma_{\tilde{\Phi}}$-strongly convex, where the constant $\sigma_{\tilde{\Phi}}$ (a.k.a. modulus) is defined as*

$$\sigma_{\tilde{\Phi}} := \min_{q,j,i} \left\{ \sigma(U^{\mathrm{w}}_q), \sigma(G^{\mathrm{b}}_i), \sigma(G^{\mathrm{d}}_{i,j}), \sigma(G^{\mathrm{c}}_i), \frac{2\kappa_j + \sigma(U^{\mathrm{v}}_j)}{2\kappa_j^2}, \frac{2\underline{\alpha}^{\mathrm{s}}_i \varrho}{M_i D_i^2} \right\}$$

where $\sigma(U^{\mathrm{w}}_q)$, $\sigma(G^{\mathrm{b}}_i)$, $\sigma(G^{\mathrm{d}}_{i,j})$, $\sigma(G^{\mathrm{c}}_i)$, $\sigma(U^{\mathrm{v}}_j)$ are the moduli of functions $-U^{\mathrm{w}}_q$, G^{b}_i, $G^{\mathrm{d}}_{i,j}$, G^{c}_i, $-U^{\mathrm{v}}_j$; the term $(2\kappa_j + \sigma(U^{\mathrm{v}}_j))/2\kappa_j^2$ is the modulus of function $(p_j \check{V}_j(p_j) - U^{\mathrm{v}}_j(p_j; V_j))$ w.r.t. p_j; and $(2\underline{\alpha}^{\mathrm{s}}_i \varrho)/(M_i D_i^2)$ is the modulus of G^{e}_i w.r.t. d_i.

Using Proposition 6 and Lemma 6, it readily follows that $\tilde{\Phi}^*$, the conjugate of the extended-valued function $\tilde{\Phi}$, is differentiable and its gradient $\nabla\tilde{\Phi}^*$ is Lipschitz continuous with constant $1/\sigma_{\tilde{\Phi}}$. That is, for any dual variables $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}^I$, we have that $\|\nabla\tilde{\Phi}^*(\boldsymbol{\pi}) - \nabla\tilde{\Phi}^*(\boldsymbol{\xi})\| \leq \frac{1}{\sigma_{\tilde{\Phi}}}\|\boldsymbol{\pi} - \boldsymbol{\xi}\|$.

To further derive the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$, let us first define $\mathbf{A}$ as an I-by-$|\mathbf{x}|$ matrix whose (i,k)-th entry is given by $\mathbf{A}_{(i,k)} := 1$ if either $\mathbf{x}_{(k)} = v_{i,j},\ \forall j$, or, $\mathbf{x}_{(k)} = w_{i,q},\ \forall q$; $\mathbf{A}_{(i,k)} := -1$ if $\mathbf{x}_{(k)} = d_i$; and $\mathbf{A}_{(i,k)} := 0$ otherwise; with $\mathbf{x}_{(k)}$ denoting the k-th entry of the optimization vector $\mathbf{x}$. Then, the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ can be rewritten as [cf. (5.22)]

$$\begin{aligned}\tilde{\mathcal{D}}(\boldsymbol{\pi}) :=& \min_{\mathbf{x}\in\tilde{\mathcal{X}}}\ \Phi(\mathbf{x}) + \sum_{i\in\mathcal{I}}\pi_i\left(\sum_{j\in\mathcal{J}} v_{i,j} + \sum_{q\in\mathcal{Q}} w_{i,q} - d_i\right)\\ =& \max_{\mathbf{x}\in\mathbf{dom}\ \tilde{\Phi}}\ -\tilde{\Phi}(\mathbf{x}) - \boldsymbol{\pi}^\top\mathbf{A}\mathbf{x} = (\tilde{\Phi})^*(-\mathbf{A}^\top\boldsymbol{\pi}).\end{aligned}$$

For any $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}^I$, it follows that

$$\begin{aligned}&\|\nabla\tilde{\mathcal{D}}(\boldsymbol{\pi}) - \nabla\tilde{\mathcal{D}}(\boldsymbol{\xi})\| = \left\|\mathbf{A}\Big(\nabla\tilde{\Phi}^*(-\mathbf{A}^\top\boldsymbol{\pi}) - \nabla\tilde{\Phi}^*(-\mathbf{A}^\top\boldsymbol{\xi})\Big)\right\|\\ \leq& \tfrac{1}{\sigma_{\tilde{\Phi}}}\|\mathbf{A}\|\cdot\|\mathbf{A}^\top(\boldsymbol{\pi} - \boldsymbol{\xi})\| \leq \tfrac{1}{\sigma_{\tilde{\Phi}}}\|\mathbf{A}\|^2\cdot\|(\boldsymbol{\pi} - \boldsymbol{\xi})\|\\ =& \tfrac{1}{\sigma_{\tilde{\Phi}}}\|\mathbf{A}^\top\mathbf{A}\|\cdot\|(\boldsymbol{\pi} - \boldsymbol{\xi})\| \leq \tfrac{1}{\sigma_{\tilde{\Phi}}}\|\mathbf{A}^\top\mathbf{A}\|_\infty\cdot\|(\boldsymbol{\pi} - \boldsymbol{\xi})\|\\ \leq& \tfrac{1}{\sigma_{\tilde{\Phi}}}\|\mathbf{A}\|_1\|\mathbf{A}\|_\infty\cdot\|(\boldsymbol{\pi} - \boldsymbol{\xi})\| \leq \tfrac{1}{\sigma_{\tilde{\Phi}}}(J+Q+1)\cdot\|(\boldsymbol{\pi} - \boldsymbol{\xi})\|\end{aligned}$$

where the last inequality holds since each row of $\mathbf{A}$ has at most $J+Q+1$ nonzero entries, and each column of $\mathbf{A}$ has at most one nonzero entry. Together with the definition of $\sigma_{\tilde{\Phi}}$, we obtain the Lipschitz constant of $\nabla\tilde{\mathcal{D}}$.

3.5.2 Proof of Proposition 4

Define the objectives of sub-problems (3.28)-(3.30) as Φ_i^1, $\Phi_{i,q}^2$ and Φ_j^3, $\forall i,j,q$, and their extended-valued functions associated with the feasible sets as $\tilde{\Phi}_i^1$, $\tilde{\Phi}_{i,q}^2$ and $\tilde{\Phi}_j^3$, $\forall i,j,q$. Let the sets $\mathcal{I}_i^{(3.28)}$, $\mathcal{I}_{i,q}^{(3.29)}$ and $\mathcal{I}_j^{(3.30)}$, $\forall i,j,q$ collect all the indexes of $\mathbf{x}$ related to (3.28)-(3.30), respectively. Using the separable nature of sub-problems (3.28)-(3.30), the sets $\mathcal{I}_i^{(3.28)}$, $\mathcal{I}_{i,q}^{(3.29)}$ and $\mathcal{I}_j^{(3.30)}$ are disjoint, and we can rewrite the dual function $\tilde{\mathcal{D}}(\boldsymbol{\pi})$ as $\tilde{\mathcal{D}}(\boldsymbol{\pi}) = \sum_{i\in\mathcal{I}}(\tilde{\Phi}_i^1)^*(-(\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_i^{(3.28)}}) + \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}(\tilde{\Phi}_{i,q}^2)^*(-(\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_{i,q}^{(3.29)}}) + \sum_{j\in\mathcal{J}}(\tilde{\Phi}_j^3)^*(-(\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_j^{(3.30)}})$, where $(\tilde{\Phi}_i^1)^*$ is the conjugate of the function $\tilde{\Phi}_i^1$, and likewise for $(\tilde{\Phi}_{i,q}^2)^*$ and $(\tilde{\Phi}_j^3)^*$. Per Lemma 6, it follows that $(\tilde{\Phi}_i^1)^*$, $(\tilde{\Phi}_{i,q}^2)^*$

and $(\tilde{\Phi}_j^3)^*$ have a Lipschitz gradient with constant $L_i^1 := \max\left\{1/\sigma(G_i^{\rm b}), 1/\sigma(G_i^{\rm c}), M_i D_i^2/2\underline{\alpha}_i^{\rm s}\varrho\right\}$, $L_{i,q}^2 := 1/\sigma(U_q^{\rm w})$, and $L_j^3 := \max_{i\in\mathcal{I}_j}\left\{1/\sigma(G_{i,j}^{\rm d}), 2\kappa_j^2/(2\kappa_j+\sigma(U_j^{\rm v}))\right\}$. Using the well-known descent lemma w.r.t. $-(\tilde{\Phi}_i^1)^*$, for any $\boldsymbol{\pi}$ and $\boldsymbol{\xi}\in\mathbb{R}^I$, we have [15]

$$\begin{aligned}(\tilde{\Phi}_i^1)^*(-(\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_i^{(3.28)}}) \geq & \left\langle \nabla(\tilde{\Phi}_i^1)^*(-(\mathbf{A}^\top\boldsymbol{\xi})_{\mathcal{I}_i^{(3.28)}}), (\mathbf{A}^\top\boldsymbol{\xi})_{\mathcal{I}_i^{(3.28)}} - (\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_i^{(3.28)}} \right\rangle \\ & + (\tilde{\Phi}_i^1)^*(-(\mathbf{A}^\top\boldsymbol{\xi})_{\mathcal{I}_i^{(3.28)}}) - \frac{L_i^1}{2}\|(\mathbf{A}^\top\boldsymbol{\xi})_{\mathcal{I}_i^{(3.28)}} - (\mathbf{A}^\top\boldsymbol{\pi})_{\mathcal{I}_i^{(3.28)}}\|^2. \end{aligned} \tag{3.37}$$

Likewise, we can obtain the descent lemma w.r.t. $-(\tilde{\Phi}_{i,q}^2)^*$ by replacing $\mathcal{I}_i^{(3.28)}$ with $\mathcal{I}_{i,q}^{(3.29)}$ and L_i^1 with $L_{i,q}^2$ in (3.37), as well as that w.r.t. $-(\tilde{\Phi}_j^3)^*$ by replacing $\mathcal{I}_i^{(3.28)}$ with $\mathcal{I}_j^{(3.30)}$ and L_i^1 with L_j^3 in (3.37). Then summing the inequalities in (3.37) as well as its counterparts for $(\tilde{\Phi}_{i,q}^2)^*$ and $(\tilde{\Phi}_j^3)^*$ over i, j, q, and using the definition of matrix $\mathbf{A}$, we can arrive at

$$-\tilde{\mathcal{D}}(\boldsymbol{\pi}) \leq -\tilde{\mathcal{D}}(\boldsymbol{\xi}) - \langle\nabla\tilde{\mathcal{D}}(\boldsymbol{\xi}), \boldsymbol{\pi}-\boldsymbol{\xi}\rangle + \frac{1}{2}(\boldsymbol{\pi}-\boldsymbol{\xi})^\top\boldsymbol{\Lambda}(\boldsymbol{\pi}-\boldsymbol{\xi}) \tag{3.38}$$

where $\boldsymbol{\Lambda}$ is a positive definite diagonal matrix with element

$$\begin{aligned}\Lambda_{ii} = & \textstyle\sum_{j\in\mathcal{J}} \max_{i\in\mathcal{I}_j}\left\{\frac{1}{\sigma(G_{i,j}^{\rm d})}, \frac{2\kappa_j^2}{2\kappa_j+\sigma(U_j^{\rm v})}\right\} \\ & + \textstyle\sum_{q\in\mathcal{Q}} \frac{1}{\sigma(U_q^{\rm w})} + \max\left\{\frac{1}{\sigma(G_i^{\rm b})}, \frac{1}{\sigma(G_i^{\rm c})}, \frac{M_i D_i^2}{2\underline{\alpha}_i^{\rm s}\varrho}\right\}.\end{aligned}$$

Using the weighted descent lemma in (3.38), the argument stated in Proposition 4 follows; see, e.g., [12, Theorem 4.4] for details.

3.5.3 Proof of Lemma 7

Along the lines in [54, Theorem 4.12], we consider a finite-state ergodic Markov chain with positive recurrence. This guarantees the existence of a sequence of finite random return times $1 = T_1 < \cdots < T_n < T_{n+1} < \infty,\ \forall n$, such that $\boldsymbol{s}_t$ visits the initial state $\boldsymbol{s}_1$ for n-th time at slot T_n. With ΔT_n denoting the n-th interval $[T_n, T_{n+1}-1]$, the sequence $\{\Delta T_n, \forall n\}$ is i.i.d. [10] and $\mathbb{E}[\Delta T_n]$ is bounded. In addition, we assume that the moment $\mathbb{E}[\Delta T_n^2]$ is finite too.

Squaring the update in (3.23a) yields

$$\begin{aligned}(\lambda_{i,t+1}^{\rm b})^2 = & (\lambda_{i,t}^{\rm b})^2 + 2\mu\lambda_{i,t}^{\rm b}P_{i,t}^{\rm b} + \mu^2(P_{i,t}^{\rm b})^2 \\ \leq & (\lambda_{i,t}^{\rm b})^2 + 2\mu\lambda_{i,t}^{\rm b}P_{i,t}^{\rm b} + \mu^2(\max_i\{\overline{P}_i^{\rm b}, -\underline{P}_i^{\rm b}\})^2\end{aligned}$$

where the last inequality follows from constraints (3.15d). Defining $P^{\mathrm{b}}_{\max}:=(\max_i\{\overline{P}^{\mathrm{b}}_i,-\underline{P}^{\mathrm{b}}_i\})^2$ and $\mathcal{T}_n:=\{T_n,\ldots,T_{n+1}-1\}$, and summing across $t\in\mathcal{T}_n$ leads to

$$\begin{aligned}(\lambda^{\mathrm{b}}_{i,T_{n+1}})^2 \leq & (\lambda^{\mathrm{b}}_{i,T_n})^2+\sum_{t\in\mathcal{T}_n}2\mu\lambda^{\mathrm{b}}_{i,t}P^{\mathrm{b}}_{i,t}+\Delta T_n\mu^2P^{\mathrm{b}}_{\max}\\ \overset{(a)}{\leq} & (\lambda^{\mathrm{b}}_{i,T_n})^2+2\mu\sum_{t\in\mathcal{T}_n}\lambda^{\mathrm{b}}_{i,T_n}P^{\mathrm{b}}_{i,t}+\Delta T_n\mu^2P^{\mathrm{b}}_{\max}+2\mu^2\sum_{t\in\mathcal{T}_n}(t-T_n+1)P^{\mathrm{b}}_{\max}\\ = & (\lambda^{\mathrm{b}}_{i,T_n})^2+2\mu\lambda^{\mathrm{b}}_{i,T_n}\sum_{t\in\mathcal{T}_n}P^{\mathrm{b}}_{i,t}+\mu^2\Delta T_n^2P^{\mathrm{b}}_{\max}\end{aligned}$$

where the inequality (a) is due to the amount of (dis)charging, which is bounded per slot so that $\lambda^{\mathrm{b}}_{i,T_n}=\lambda^{\mathrm{b}}_{i,t}+\sum_{T_n}^{T_{n+1}-1}P^{\mathrm{b}}_{i,t}$, and $\lambda^{\mathrm{b}}_{i,T_n}+(t-T_n+1)\underline{P}^{\mathrm{b}}_i\leq\lambda^{\mathrm{b}}_{i,t}\leq\lambda^{\mathrm{b}}_{i,T_n}+(t-T_n+1)\overline{P}^{\mathrm{b}}_i$.

Likewise, squaring the update in (3.23b)-(3.23c) yields

$$(\lambda^{\mathrm{dc}}_{i,q,T_{n+1}})^2\leq(\lambda^{\mathrm{dc}}_{i,q,T_n})^2+2\mu\lambda^{\mathrm{dc}}_{i,q,T_n}\sum_{t\in\mathcal{T}_n}\Big(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}-w_{i,q,t}\Big)+\mu^2\Delta T_n^2\Big(\max\left\{M_iD_i,\sum_{j\in\mathcal{J}}B_{j,i}\right\}\Big)^2$$

and

$$(\lambda^{\mathrm{mn}}_{j,q,T_{n+1}})^2\leq(\lambda^{\mathrm{mn}}_{j,q,T_n})^2+2\mu\lambda^{\mathrm{mn}}_{j,q,T_n}\sum_{t\in\mathcal{T}_n}\Big(W_{j,q,t}-\sum_{i\in\mathcal{I}}\tilde{w}_{i,j,q,t}\Big)+\mu^2\Delta T_n^2\Big(\max\left\{\overline{W}_{j,q},\sum_{i\in\mathcal{I}}B_{j,i}\right\}\Big)^2$$

where these inequalities come from (4.21e), (5.7b)-(4.21j), and the fact that $W_{j,q}\leq\overline{W}_{j,q}$.

Upon defining the Lyapunov drift as

$$\begin{aligned}\Delta(\boldsymbol{\varpi}_{T_n}):=\frac{1}{2}\Bigg[&\sum_{i\in\mathcal{I}}(\lambda^{\mathrm{b}}_{i,T_{n+1}})^2-\sum_{i\in\mathcal{I}}(\lambda^{\mathrm{b}}_{i,T_n})^2+\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}(\lambda^{\mathrm{dc}}_{i,q,T_{n+1}})^2\\ &-\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}(\lambda^{\mathrm{dc}}_{i,q,T_n})^2+\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}(\lambda^{\mathrm{mn}}_{j,q,T_{n+1}})^2-\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}(\lambda^{\mathrm{mn}}_{j,q,T_n})^2\Bigg]\end{aligned}$$

we have

$$\begin{aligned}\Delta(\boldsymbol{\varpi}_{T_n})\leq\mu\sum_{t\in\mathcal{T}_n}\Bigg(&\sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{dc}}_{i,q,T_n}\Bigg(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t}-w_{i,q,t}\Bigg)\\ &+\sum_{i\in\mathcal{I}}\lambda^{\mathrm{b}}_{i,T_n}P^{\mathrm{b}}_{i,t}+\sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{mn}}_{j,q,T_n}\Big(W_{j,q,t}-\sum_{i\in\mathcal{I}}\tilde{w}_{i,j,q,t}\Big)\Bigg)+\mu^2\Delta T_n^2M.\end{aligned}$$

Adding $\mu\sum_{t\in\mathcal{T}_n}\Psi_t(\mathbf{x}_t)$ to both sides, and taking the expectation conditioned on $\boldsymbol{\varpi}_{T_n}$ over the duration of the renewal interval ΔT_n as well as the random state $\boldsymbol{s}_t$ on each slot of this period, we

arrive at

$$
\begin{aligned}
&\mathbb{E}[\Delta(\boldsymbol{\varpi}_{T_n})] + \mu\mathbb{E}\left[\sum_{t\in\mathcal{T}_n}\Psi_t(\mathbf{x}_t)\right] \\
\leq&\mu\mathbb{E}\Bigg[\sum_{t\in\mathcal{T}_n}\Bigg(\Psi_t(\mathbf{x}_t) + \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{dc}}_{i,q,T_n}\left(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t} - w_{i,q,t}\right) \\
&+ \sum_{i\in\mathcal{I}}\lambda^{\mathrm{b}}_{i,T_n}P^{\mathrm{b}}_{i,t} + \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{mn}}_{j,q,T_n}\left(W_{j,q,t} - \sum_{i\in\mathcal{I}}\tilde{w}_{i,j,q,t}\right)\Bigg)\Bigg] + \mu^2 M\mathbb{E}[\Delta T_n^2] \\
\overset{(a)}{=}&\mu\mathbb{E}[\Delta T_n]\mathbb{E}\Bigg[\Bigg(\Psi_t(\mathbf{x}_t) + \sum_{i\in\mathcal{I}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{dc}}_{i,q,T_n}\left(\sum_{j\in\mathcal{J}}\tilde{w}_{i,j,q,t} - w_{i,q,t}\right) \\
&+ \sum_{i\in\mathcal{I}}\lambda^{\mathrm{b}}_{i,T_n}P^{\mathrm{b}}_{i,t} + \sum_{j\in\mathcal{J}}\sum_{q\in\mathcal{Q}}\lambda^{\mathrm{mn}}_{j,q,T_n}\left(W_{j,q,t} - \sum_{i\in\mathcal{I}}\tilde{w}_{i,j,q,t}\right)\Bigg)\Bigg] + \mu^2 M\mathbb{E}[\Delta T_n^2] \\
\overset{(b)}{=}&\mu\mathbb{E}[\Delta T_n]\mathcal{L}(\mathbf{x}(\boldsymbol{\varpi}_{T_n}), \boldsymbol{\varpi}_{T_n}) + \mu^2 M\mathbb{E}[\Delta T_n^2] \\
\overset{(c)}{\leq}&\mu\mathbb{E}[\Delta T_n]\mathcal{D}(\boldsymbol{\varpi}_{T_n}) + \mu^2 M\mathbb{E}[\Delta T_n^2] \\
\overset{(d)}{\leq}&\mu\mathbb{E}[\Delta T_n]\tilde{\Psi}^* + \mu^2 M\mathbb{E}[\Delta T_n^2] \overset{(e)}{\leq} \mu\mathbb{E}[\Delta T_n]\Psi^* + \mu^2 M\mathbb{E}[\Delta T_n^2]
\end{aligned}
$$

where equality (a) is due to the basic renewal theorem [10], with the first and third expectations over ΔT_n, and the second one over the random state $\boldsymbol{s}_t$; equality (b) follows from the definition of the Lagrangian in (5.11) with $\mathbf{x}(\boldsymbol{\varpi}_{T_n})$ denoting the optimal primal variables given by (3.24); equality (c) comes from the definition of the dual function; inequality (d) follows from the weak duality; and inequality (e) holds since (5.9) is a relaxation of (5.7).

Summing over all $n = 1, 2, \ldots, N$, we then have

$$
\begin{aligned}
&\textstyle\sum_{n=1}^{N}\mathbb{E}[\Delta(\boldsymbol{\varpi}_{T_n})] + \mu\sum_{n=1}^{N}\mathbb{E}\left[\sum_{t\in\mathcal{T}_n}\Psi_t(\mathbf{x}_t)\right] \\
\leq&\textstyle\sum_{n=1}^{N}\mu\mathbb{E}[\Delta T_n]\tilde{\Psi}^* + \mu^2 NM\mathbb{E}[\Delta T_n^2],
\end{aligned}
$$

which dividing by $\mu N\mathbb{E}[\Delta T_n]$ and letting $N \to \infty$ leads to

$$
\begin{aligned}
&\lim_{T\to\infty}\tfrac{1}{T}\textstyle\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi\left(\mathbf{x}_t(\boldsymbol{\varpi}_t)\right)\right] \\
=&\lim_{N\to\infty}\textstyle\sum_{n=1}^{N}\frac{1}{N\mathbb{E}[\Delta T_n]}\mathbb{E}\left[\sum_{t\in\mathcal{T}_n}\Psi_t(\mathbf{x}_t)\right] \overset{(a)}{\leq} \Psi^* + \mu M\frac{\mathbb{E}[\Delta T_n^2]}{\mathbb{E}[\Delta T_n]}
\end{aligned}
$$

where $\overset{(a)}{\leq}$ holds true because the multipliers are bounded.

Chapter 4

Robust Resource Allocation over Data Center Networks

4.1 System Models

Consider a network with geographically distributed MNs $\mathcal{J} := \{1, 2, \ldots, J\}$ and DCs $\mathcal{I} := \{1, 2, \ldots, I\}$.

4.1.1 Network and workload models

In general, DC workloads are either delay-sensitive (interactive) or delay-tolerant [85]. The interactive workloads entailing real-time user requests must be attended to immediately; e.g., instant messages and voice services. In contrast, delay-tolerant workloads such as system updates and data backups are deferrable within a given time interval. This flexibility of delay-tolerant loads enables opportunistic workload management that can be adaptive to the time-varying energy prices and renewables.

Consider a discrete-time scheduling horizon[1] $\mathcal{T} := \{1, \ldots, T\}$. For interactive workloads, let A_j^t denote the arrival rate of service requests at MN j, and a_{ji}^t the workload directed from node j to DC i over slot t. For delay-tolerant workloads, let $\mathcal{Q}_j$ denote the jobs collected by node j, and

[1]For convenience, the slot duration is normalized to unity; thus, the terms "energy" and "power" will be interchangeably used throughout this chapter.

$\mathcal{Q} := \bigcup_{j=1}^{J} \mathcal{Q}_j$ with $\mathcal{Q}_i \bigcap \mathcal{Q}_j = \emptyset, \forall i \neq j$, representing the set of all delay-tolerant jobs. The qth delay-tolerant job can be specified by its total demand B_q, and active interval $\mathcal{T}_q := \{S_q, \ldots, E_q\}$, with S_q and E_q denoting the start- and end-time slots. Let $\tilde{b}_{i,q}^t$ denote the amount of qth delay-tolerant job routed from[2] its MN to DC i, and $b_{i,q}^t$ the amount processed by DC i over slot t, respectively; and L_{ji}^t denote the link bandwidth from node j to DC i at time t. As shown in Fig. 4.1, these quantities must satisfy the following constraints:

$$\sum_{i=1}^{I} a_{ji}^t = A_j^t, \ \forall j \in \mathcal{J}, \ t \in \mathcal{T} \tag{4.1}$$

$$\sum_{t=S_q}^{E_q} \sum_{i=1}^{I} \tilde{b}_{i,q}^t = B_q, \ \forall q \in \mathcal{Q} \tag{4.2}$$

$$a_{ji}^t + \sum_{q \in \mathcal{Q}_j} \tilde{b}_{i,q}^t \leq L_{ji}^t, \ \forall i \in \mathcal{I}, \ j \in \mathcal{J}, \ t \in \mathcal{T} \tag{4.3}$$

where (4.1) ensures that interactive workloads are dispatched once arrived; (4.2) requires routing each delay-tolerant job before its deadline; and (5.4) captures the bandwidth limitation of data transfer. Clearly, $L_{ji}^t = 0$ if load transfer from node j to DC i is prohibited; e.g., when MN j is not physically linked with DC i.

In each DC, interactive workloads are processed immediately, while delay-tolerant workloads are deferrable. The unserved portion of delay-tolerant workloads are buffered in separate queues obeying following dynamic recursions

$$Q_{i,q}^{t+1} = Q_{i,q}^t - b_{i,q}^t + \tilde{b}_{i,q}^t, \ \ \forall i \in \mathcal{I}, t \in \mathcal{T}_q, q \in \mathcal{Q} \tag{4.4}$$

where $Q_{i,q}^t$ is the queue length of qth delay-tolerant job in DC i at the beginning of slot t. For the deadline completion requirements, queue length $Q_{i,q}^t$ must satisfy

$$Q_{i,q}^t \geq 0, \quad Q_{i,q}^{S_q} = Q_{i,q}^{E_q+1} = 0. \tag{4.5}$$

The total IT demand of DC i in slot t, is thus given by

$$d_i^t = \sum_{j=1}^{J} a_{ji}^t + \sum_{q \in \mathcal{Q}} b_{i,q}^t, \ \forall t \in \mathcal{T}. \tag{4.6}$$

[2]Workload distribution delay is ignored here.

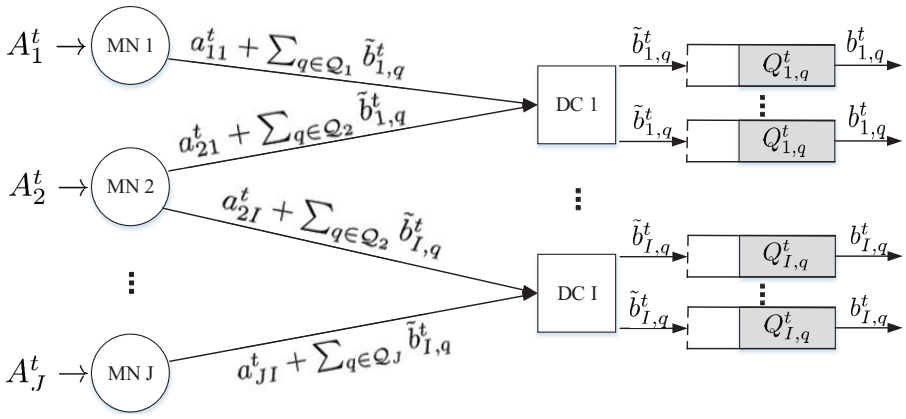

Figure 4.1: A workload distribution diagram.

The aforementioned IT system mainly deals with the DC network workload balancing tasks by exploiting the heterogeneous server and bandwidth resources in the cloud. The control variables therein are $\{d_i^t, a_{ji}^t, b_{i,q}^t, \tilde{b}_{i,q}^t\}$, under constraints (4.1)–(4.6). In addition, the IT system also closely connects with the underlying power infrastructure through a power supply and demand relationship, which is instructive to detail in the next two subsections.

4.1.2 Power demand model

The power consumption of a DC generally comes from various sources, but mainly from the running servers and cooling systems [77].

Suppose that each DC i has a set of $\overline{M}_i$ homogeneous servers, so the number of active servers m_i^t at time t should be in the range $\underline{M}_i \le m_i^t \le \overline{M}_i$, where $\underline{M}_i$ stands for the minimum number of servers required for providing basic services. The consumption of each server can be generally modeled as a function of its running speed [81]

$$P_{i,s}(s_i^t) = \overline{P}_{i,s}\left(\varrho(s_i^t)^\upsilon + 1 - \varrho\right), \quad \varrho \in [0,1]$$

where $\overline{P}_{i,s}$ denotes the peak power consumption of a server in DC i; $s_i^t \in [0,1]$ is its actual speed (a.k.a. CPU usage); and parameter υ is typically around 2 for state-of-the-art servers [78]. Clearly, the fraction of peak consumption $1-\varrho$ denotes the power consumed in the idle state (i.e., $s_i^t = 0$). When the server is in its highest speed $s_i^t = 1$, the actual consumption is $\overline{P}_{i,s}$.

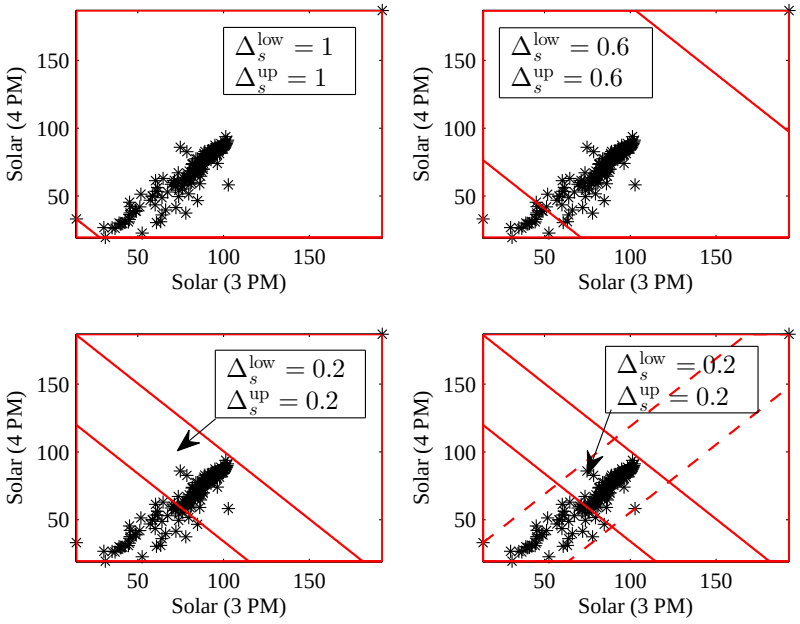

Figure 4.2: A diagram of uncertainty sets of the CAISO solar generations.

Due to the convexity of $P_{i,s}(s_i^t)$, it readily follows that given a total IT demand d_i^t, uniform allocation of the workloads to each server is most energy efficient [85]. Accordingly, each server is running at a speed $d_i^t/(m_i^t D_i)$ with D_i denoting the server capacity under the required service level agreement (SLA), and the total power consumption in DC i becomes

$$P_{i,IT}(d_i^t, m_i^t) = \frac{\varrho \overline{P}_{i,s} {d_i^t}^2}{m_i^t D_i^2} + (1-\varrho)\overline{P}_{i,s} m_i^t. \tag{4.7}$$

Since the number of servers is very large, m_i^t can be relaxed to be a positive real number for simplicity [46].

Along with the increasing density of IT equipment in DCs, a considerable amount of electricity is consumed by the cooling system that generally operates in two modes [9, 48]: outside-air and chilled-water cooling. The energy usage of *outside-air cooling* is mainly the power consumed by blowers, which can be approximated as a cubic function of the blower speed [91]. As the blower speed under tight control is proportional to the IT consumption $P_{i,IT}$, the outside-air cooling power consumption can be modeled as a convex function of $P_{i,IT}$, namely

$$\mathcal{F}_{i,a}^t(P_{i,IT}) = \kappa_i^t (P_{i,IT})^3, \;\; 0 \le P_{i,IT} \le \overline{P}_{i,a}^t \tag{4.8}$$

where $\kappa_i^t > 0$ depends on the temperature difference between the (hot) exhausting air temperature $T_{i,RA}^t$ from the IT racks and the outside air temperature $T_{i,OA}^t$ around DC i at time slot t. The capacity of outside-air cooling in (4.8) can be modeled as $\overline{P}_{i,a}^t = C(T_{i,RA}^t - T_{i,OA}^t)$, with $C > 0$ proportional to the maximal outside air flow rate.

The *chilled-water cooling* model here is established on the actual measurement of a practical chiller for which the power consumption can be approximated as [59]

$$\mathcal{F}_{i,w}(P_{i,IT}) = \gamma P_{i,IT} \tag{4.9}$$

where $P_{i,IT}$ is again the IT power consumption in (4.7) and $\gamma > 0$ is a constant depending on the specific chiller characteristics.

Due to different cooling efficiencies and capacities of the two approaches, for a given $P_{i,IT}$, there is an optimal allocation between outside-air cooling and chiller cooling. Let $P_{i,w}^t$ and $P_{i,a}^t$ denote the amounts of IT power consumption allocated for chiller and outside-air cooling, respectively. With (4.8) and (4.9), the optimal cooling power consumption is [48]

$$\begin{aligned}\mathcal{F}_i^t(P_{i,IT}) = & \min_{\substack{0 \le P_{i,a}^t \le \overline{P}_{i,a}^t \\ P_{i,a}^t + P_{i,w}^t = P_{i,IT}}} \mathcal{F}_{i,w}(P_{i,w}^t) + \mathcal{F}_{i,a}^t(P_{i,a}^t) \\ = & \min_{0 \le P_{i,a}^t \le \overline{P}_{i,a}^t} \gamma[P_{i,IT} - P_{i,a}^t]^+ + \kappa_i^t (P_{i,a}^t)^3. \end{aligned} \tag{4.10}$$

With a temperature-dependent threshold $P_{i,TH}^t := \min\{\overline{P}_{i,a}^t, \sqrt{\gamma/3\kappa_i^t}\}$, (4.10) admits a closed-form solution

$$\mathcal{F}_i^t(P_{i,IT}) = \begin{cases} \kappa_i^t (P_{i,IT})^3, & P_{i,IT} \le P_{i,TH}^t \\ \kappa_i^t (P_{i,TH}^t)^3 + \gamma(P_{i,IT} - P_{i,TH}^t), & \text{otherwise.} \end{cases}$$

For notational convenience, let $P_i^t(d_i^t, m_i^t)$ include the server and cooling consumptions in DC i per slot t as $P_i^t(d_i^t, m_i^t) := \mathcal{F}_i^t(d_i^t, m_i^t) + P_{i,IT}(d_i^t, m_i^t)$, and consider the following lemma.

Lemma 10. *Function $P_i^t(d_i^t, m_i^t)$ is jointly convex in $\{d_i^t, m_i^t\}$.*

Proof. See Appendix A. □

4.1.3 Power supply model

A rapidly increasing use of microgrids characterizes the transformative change from our aging power grid to a smart grid over the last decade [43]. While the traditional geographical workload balancing operates separately from the local power balancing, recently advocated dynamic pricing and demand response programs motivate the interactions between them. In this context, we consider each DC to be supplied by a RES-integrated microgrid consisting of a conventional generator (CG) (e.g., a fuel generator), an on-site renewable generator (RG) (e.g., a wind or solar generator), and an energy storage unit (e.g., a battery).

Let $P_{i,g}^t$ denote the energy output of the CG in DC i per slot t, which is upper bounded by $\overline{P}_{i,g}$; that is,

$$0 \leq P_{i,g}^t \leq \overline{P}_{i,g},\ \forall i \in \mathcal{I}, t \in \mathcal{T}. \tag{4.11}$$

The change of CG energy output in two consecutive slots is bounded by the following so-termed ramping constraints:

$$P_{i,g}^t - P_{i,g}^{t-1} \leq R_i^{\mathrm{up}},\ \ P_{i,g}^{t-1} - P_{i,g}^t \leq R_i^{\mathrm{dw}},\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.12}$$

where R_i^{up} and R_i^{dw} are the ramping-up and ramping-down limits of CG at DC i.

Consider now the RES vector $\mathbf{e}_i := [E_i^1, \ldots, E_i^T]'$ generated at DC i across all slots. Due to the unpredictable and intermittent nature of RES, $\mathbf{e}_i$ is unknown a priori. In general, uncertain quantities can be modeled by postulating either an underlying probability distribution or an uncertainty region. Probability distributions (possibly mixed discrete/continuous) of the RES generation are seldom available in practice. Although (non-)parametric approaches can be used to learn these distributions, the processes can be very complicated due to the spatio-temporal correlations incurred by various meteorological factors [90]. On the contrary, the proposed method of postulating an uncertainty region provides the decision maker with ranges instead of point forecasts, which is essentially a distribution-free deterministic set and *robust* to prediction errors.

The actual RES generation $\mathbf{e}_i$ is assumed to lie in an uncertainty set $\mathcal{E}_i$, which can be obtained via forecasting or inference using historical measurements. In particular, the following polyhedral

uncertainty set is considered (see also [17, 89])

$$\mathcal{E}_i := \Bigg\{ \mathbf{e}_i \mid \underline{E}_i^t \leq E_i^t \leq \overline{E}_i^t,\ \mathcal{T} = \bigcup_{s=1}^{S} \mathcal{T}_{i,s},$$

$$\Delta_s^{\text{low}} \underline{E}_{\mathcal{T}_{i,s}} + (1 - \Delta_s^{\text{low}}) E_{\mathcal{T}_{i,s}}^{\text{avg}} \leq \sum_{t \in \mathcal{T}_{i,s}} E_i^t \leq (1 - \Delta_s^{\text{up}}) E_{\mathcal{T}_{i,s}}^{\text{avg}} + \Delta_s^{\text{up}} \overline{E}_{\mathcal{T}_{i,s}} \Bigg\} \quad (4.13)$$

where $\underline{E}_i^t$ ($\overline{E}_i^t$) denotes the lower (upper) bound on the actual E_i^t; $\mathcal{T}$ is partitioned into consecutive but non-overlapping sub-horizons $\mathcal{T}_{i,s}$, $s = 1, \ldots, S$; $E_{\mathcal{T}_{i,s}}^{\text{avg}}$ is the sample average of total renewables $\sum_{t \in \mathcal{T}_{i,s}} E_i^t$; the total renewables over $\mathcal{T}_{i,s}$ are bounded by $\underline{E}_{\mathcal{T}_{i,s}}$ and $\overline{E}_{\mathcal{T}_{i,s}}$; and the parameter $\Delta_s^{\text{low}}(\Delta_s^{\text{up}}) \in [0, 1]$ represents the *level of robustness* for the lower (upper) bound of the sth sub-horizon. Note that all the aforementioned statistics can be directly obtained using real RES generations from public sources.

Clearly, parameters Δ_s^{low}, Δ_s^{up} trade off robustness for conservatism of the resultant solutions. Based on the CAISO solar generations during Mar. 1 – Oct. 30, 2012 [6], examples are given in Fig. 4.2 for $|\mathcal{T}_{i,s}| = 2$; e.g., each sub-horizon consists of two time slots. In Fig. 4.2, Figs. 3(a)-3(c) are for $\mathcal{E}_i$, and Fig. 3(d) is for $\check{\mathcal{E}}_i$. Points ($*$) denote the generation samples from historical data. Red lines represent the boundaries of the polyhedral uncertainty sets (4.13) and (4.14). When the robustness levels Δ_s^{low} and Δ_s^{up} are high, the uncertainty set is large, which includes most of historical samples thus usually leading to conservative solutions when the degree of uncertainty is high; if Δ_s^{low} and Δ_s^{up} are low, some of samples are excluded from the uncertainty set, which reduces the robustness of the resultant solutions.

Here it is also instructive to point out that by further capturing the maximum variation of RES over two consecutive slots $\overline{\partial E}_i^t$, a more accurate polyhedral set can be written as (see Fig. 4.2(d))

$$\check{\mathcal{E}}_i := \Bigg\{ \mathbf{e}_i \mid \underline{E}_i^t \leq E_i^t \leq \overline{E}_i^t,\ |E_i^t - E_i^{t-1}| \leq \overline{\partial E}_i^t,\ \mathcal{T} = \bigcup_{s=1}^{S} \mathcal{T}_{i,s},$$

$$\Delta_s^{\text{low}} \underline{E}_{\mathcal{T}_{i,s}} + (1 - \Delta_s^{\text{low}}) E_{\mathcal{T}_{i,s}}^{\text{avg}} \leq \sum_{t \in \mathcal{T}_{i,s}} E_i^t \leq (1 - \Delta_s^{\text{up}}) E_{\mathcal{T}_{i,s}}^{\text{avg}} + \Delta_s^{\text{up}} \overline{E}_{\mathcal{T}_{i,s}} \Bigg\}. \quad (4.14)$$

This modification allows a decision maker to reduce the uncertainty region, without missing many potential samples. It makes sense intuitively since in a short time scale (e.g., 15 min, or, an hour), the RES are highly correlated over successive slots. Although the aforementioned practical models only capture RES uncertainty across the scheduling horizons per DC, our proposed approach could be easily extended to include joint spatio-temporal uncertainty models.

To mitigate the variability of RES, energy storage devices are recently considered so as to store the surplus renewables for later shortage [34]. We consider a storage unit with finite capacity $\overline{C}_i$, and let C_i^0 and C_i^t denote the initial energy level of the storage unit in DC i at the beginning of time slot t. Since storage devices become unreliable with high depth-of-discharge[3] (DoD), a nonzero minimum level $\underline{C}_i$ can avoid high DoD. Such a level could also support the DC operation in the event of a grid outage. Let $P_{i,\mathrm{ch}}^t$ and $P_{i,\mathrm{dis}}^t$ denote the amounts of power charging and discharging the storage unit (battery) in DC i at slot t. Due to AC/DC power conversion during the (dis-)charging process, the power conversion losses need to be accounted for by the (dis-)charging efficiency $\delta \in (0, 1]$. In addition, dissipation losses due to battery energy leakages are captured by the efficiency coefficient $\eta \in (0, 1]$, which renders a decreasing energy level even if there is no (dis-)charging operation. In short, the energy storage unit can be compactly described as

$$C_i^{t+1} = \eta C_i^t + \delta P_{i,\mathrm{ch}}^t - \frac{P_{i,\mathrm{dis}}^t}{\delta}, \ \ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.15}$$

$$\underline{C}_i \leq C_i^t \leq \overline{C}_i, \ \ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.16}$$

$$0 \leq P_{i,\mathrm{dis}}^t \leq \overline{P}_{i,\mathrm{dis}}; \ 0 \leq P_{i,\mathrm{ch}}^t \leq \overline{P}_{i,\mathrm{ch}}, \ \ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.17}$$

where the bounds $\overline{P}_{i,\mathrm{dis}}$ and $\overline{P}_{i,\mathrm{ch}}$ on (dis-)charging amounts are dictated by physical limits.

Let $P_{i,\mathrm{out}}^t$ denote the total energy consumption of DC i per slot t including the IT operating consumption, cooling power consumption, and battery charged power; that is, [cf. Lemma 10]

$$P_{i,\mathrm{out}}^t = P_i^t + P_{i,\mathrm{ch}}^t. \tag{4.18}$$

Likewise, the total energy supply $P_{i,\mathrm{in}}^t$ in DC i per slot t is given by

$$P_{i,\mathrm{in}}^t = P_{i,g}^t + E_i^t + P_{i,\mathrm{dis}}^t. \tag{4.19}$$

[3]DoD is the percentage of maximum charge removed during a discharge cycle.

Under constraints (4.11)-(4.19), the power supply optimization variables are CG and battery power amounts $\{P_{i,g}^t, P_{i,\mathrm{dis}}^t, P_{i,\mathrm{ch}}^t, C_i^t\}$.

4.1.4 Cost-revenue model

In addition to the internal energy resources (namely, CG, RG, storage unit), DCs can resort to the main grid market in an on-demand manner. With a two-way energy trading facility, each DC can buy energy from external energy markets in the case of a deficit ($P_{i,\mathrm{out}}^t > P_{i,\mathrm{in}}^t$), or, sell energy to the markets in the case of a surplus ($P_{i,\mathrm{out}}^t < P_{i,\mathrm{in}}^t$). Clearly, the shortage energy that needs to be purchased by the DC is $[P_{i,\mathrm{out}}^t - P_{i,\mathrm{in}}^t]^+$; while the surplus energy that can be sold is $[P_{i,\mathrm{in}}^t - P_{i,\mathrm{out}}^t]^+$. Note that both the shortage and surplus energies are non-negative, and at most one of them is positive at any slot t.

Suppose that the energy can be purchased from the wholesale electricity market around DC i in period t at price α_i^t, while the energy is sold at price β_i^t. Notwithstanding, we shall always set $\alpha_i^t \geq \beta_i^t$ to avoid less relevant buy-and-sell activities of the DC for profit. For DC i, the *worst-case transaction cost* for the whole scheduling horizon is defined as

$$G_i(\{P_{i,\mathrm{out}}^t\}, \{P_{i,\mathrm{in}}^t\}) := \max_{\mathbf{e}_i \in \mathcal{E}_i} \sum_{t=1}^{T} \alpha_i^t [P_{i,\mathrm{out}}^t - P_{i,\mathrm{in}}^t]^+ - \beta_i^t [P_{i,\mathrm{in}}^t - P_{i,\mathrm{out}}^t]^+ \tag{4.20}$$

which is the point-wise maximum over any realization of the random RES generation in the uncertainty set.

In addition, let function $G_{C_i}(P_{i,g}^t)$ denote the cost of CG at DC i in slot t, which is convex piecewise linear or smooth quadratic [79]. The revenue considered here comes from processing delay-tolerant workloads. Specifically, for the qth job, the revenue earned per slot t can be generally modeled as a concave function $U_q^t(b_{i,q}^t)$, which reflects the diminishing marginal sensitivity of end users to the increasing gains.

4.2 Robust Workload and Energy Management

Based on the practical models in Section II, we pursue in this section a robust workload and energy management approach for the considered DC network. Over the scheduling horizon $\mathcal{T}$, the

system operator per MN performs an (e.g. hour-) ahead-of-time schedule to optimize workloads routing $\{a_{ji}^t, \tilde{b}_{i,q}^t\}$, while the system operator in each DC optimizes servers and workloads scheduling $\{m_i^t, b_{i,q}^t\}$, CG generation $\{P_{i,g}^t\}$, and battery (dis-)charging energy $\{P_{i,\mathrm{ch}}^t, P_{i,\mathrm{dis}}^t\}$. The optimal management strategy minimizes the worst-case net cost $\tilde{\Psi}$, which includes the worst-case transaction cost, the CG cost and the revenue of delay-tolerant workloads, subject to DC operating and power supply constraints. Note that the worst-case net cost here is the maximum net cost for any realization of the random RES generation in the uncertainty set.

With $\tilde{\mathbf{x}}$ collecting all the optimization variables $\{a_{ji}^t, b_{i,q}^t, \tilde{b}_{i,q}^t, d_i^t, m_i^t, P_{i,g}^t, P_{i,\mathrm{ch}}^t, P_{i,\mathrm{dis}}^t, C_i^t\}$, we wish to solve

$$\tilde{\Psi}^* := \min_{\tilde{\mathbf{x}}} \sum_{i=1}^{I} G_i(\{P_{i,\mathrm{out}}^t\}, \{P_{i,\mathrm{in}}^t\}) + \sum_{t=1}^{T}\sum_{i=1}^{I}\left(G_{C_i}(P_{i,g}^t) - \sum_{q\in\mathcal{Q}} U_q^t(b_{i,q}^t)\right) \tag{4.21a}$$

subject to:

$$C_i^{t+1} = \eta C_i^t + \delta P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t/\delta,\ \underline{C}_i \le C_i^t \le \overline{C}_i,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21b}$$

$$0 \le P_{i,\mathrm{dis}}^t \le \overline{P}_{i,\mathrm{dis}},\ \forall i \in \mathcal{I},\ 0 \le P_{i,\mathrm{ch}}^t \le \overline{P}_{i,\mathrm{ch}},\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21c}$$

$$0 \le P_{i,g}^t \le \overline{P}_{i,g},\ P_{i,g}^t - P_{i,g}^{t-1} \le R_i^{\mathrm{up}},\ P_{i,g}^{t-1} - P_{i,g}^t \le R_i^{\mathrm{dw}},\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21d}$$

$$\underline{M}_i \le m_i^t \le \overline{M}_i,\ 0 \le d_i^t \le m_i^t D_i,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21e}$$

$$a_{ji}^t + \textstyle\sum_{q\in\mathcal{Q}_j} \tilde{b}_{i,q}^t \le L_{ji}^t,\ \forall j \in \mathcal{J}, i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21f}$$

$$\textstyle\sum_{i=1}^{I} a_{ji}^t = A_j^t,\ \forall j \in \mathcal{J}, t \in \mathcal{T},\ \sum_{t=S_q}^{E_q}\sum_{i=1}^{I} \tilde{b}_{i,q}^t = B_q,\ \forall q \in \mathcal{Q} \tag{4.21g}$$

$$Q_{i,q}^{t+1} = Q_{i,q}^t - b_{i,q}^t + \tilde{b}_{i,q}^t,\ \forall i \in \mathcal{I}, t \in \mathcal{T}, q \in \mathcal{Q} \tag{4.21h}$$

$$P_{i,\mathrm{out}}^t = P_i^t(d_i^t, m_i^t) + P_{i,\mathrm{ch}}^t,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21i}$$

$$P_{i,\mathrm{in}}^t = P_{i,g}^t + E_i^t + P_{i,\mathrm{dis}}^t,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21j}$$

$$Q_{i,q}^{S_q} = Q_{i,q}^{E_q+1} = 0,\ Q_{i,q}^t \ge 0,\ \forall i \in \mathcal{I}, q \in \mathcal{Q}, t \in \mathcal{T}_q \tag{4.21k}$$

$$d_i^t = \textstyle\sum_{j=1}^{J} a_{ji}^t + \sum_{q\in\mathcal{Q}} b_{i,q}^t,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.21l}$$

$$0 \le b_{i,q}^t \le B_q,\ \forall t \in \mathcal{T}_q;\ b_{i,q}^t = 0,\ \forall i \in \mathcal{I}, q \in \mathcal{Q}, t \notin \mathcal{T}_q \tag{4.21m}$$

$$0 \le \tilde{b}_{i,q}^t \le B_q,\ \forall t \in \mathcal{T}_q;\ \tilde{b}_{i,q}^t = 0,\ \forall i \in \mathcal{I}, q \in \mathcal{Q}, t \notin \mathcal{T}_q \tag{4.21n}$$

$$0 \le a_{ji}^t \le A_j^t,\ \forall j \in \mathcal{J}, i \in \mathcal{I}, t \in \mathcal{T}. \tag{4.21o}$$

It is worth mentioning that thanks to the worst-case transaction cost $G_i(\{P_{i,\mathrm{out}}^t\}, \{P_{i,\mathrm{in}}^t\})$, the

objective of (5.7) has an implicit min-max form, and the RES induced randomness can be eliminated; thus, (5.7) contains only deterministic variables. However, the objective of (5.7) is to minimize a point-wise maximum function, which is generally not differentiable when the maximum is attained by more than one solution. In addition, since P_i^t is a nonlinear function with respect to $\{d_i^t, m_i^t\}$, then (4.21i) are nonlinear equality constraints representing a nonconvex feasible set [19, Chap. 4]. Thus, problem (5.7) is nonsmooth and nonconvex, which is hard to be handled by existing solvers. To turn (5.7) into a tractable form, a reformulation relying on epigraph-based relaxation is pursued next.

4.2.1 Convex reformulation

Define $\psi_i^t := (\alpha_i^t - \beta_i^t)/2$, $\phi_i^t := (\alpha_i^t + \beta_i^t)/2$, and $R_i^t = P_i^t + P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t - P_{i,g}^t$; and then rewrite (4.20) as

$$G_i(\{R_i^t\}) = \max_{\mathbf{e}_i \in \mathcal{E}_i} \sum_{t=1}^{T} \left(\psi_i^t |R_i^t - E_i^t| + \phi_i^t (R_i^t - E_i^t) \right). \tag{4.22}$$

In order to convexify the (5.7) and facilitate a distributed implementation, we define $\mathbf{x}$ collecting all the optimization variables $\{a_{ji}^t, b_{i,q}^t, \tilde{b}_{i,q}^t, m_i^t, d_i^t, R_i^t, P_{i,g}^t, P_{i,\mathrm{ch}}^t, P_{i,\mathrm{dis}}^t, C_i^t\}$, and rewrite (5.7) as (RWEM):

$$\Psi^* := \min_{\mathbf{x}} \sum_{t=1}^{T} \sum_{i=1}^{I} \left(G_{C_i}(P_{i,g}^t) - \sum_{q \in \mathcal{Q}} U_q^t(b_{i,q}^t) \right) + \sum_{i=1}^{I} G_i(\{R_i^t\}) \tag{4.23a}$$

subject to:

$$(4.21b) - (5.7c), (4.21m) - (4.21o) \tag{4.23b}$$

$$R_i^t \geq P_i^t(d_i^t, m_i^t) + P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t - P_{i,g}^t, \ \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{4.23c}$$

$$\sum_{\tau=S_q}^{E_q} \tilde{b}_{i,q}^\tau = \sum_{\tau=S_q}^{E_q} b_{i,q}^\tau, \ \sum_{\tau=S_q}^{t} \tilde{b}_{i,q}^\tau \geq \sum_{\tau=S_q}^{t} b_{i,q}^\tau, \ \forall i, q \in \mathcal{Q}, t \in [S_q, E_q - 1] \tag{4.23d}$$

$$d_i^t = \sum_{j=1}^{J} a_{ji}^t + \sum_{q \in \mathcal{Q}} b_{i,q}^t, \ \forall i \in \mathcal{I}, t \in \mathcal{T}. \tag{4.23e}$$

Convexity of the worst-case net cost Ψ is established in the following proposition.

Proposition 7. *If $\alpha_i^t \geq \beta_i^t$ holds for all $i \in \mathcal{I}$ and $t \in \mathcal{T}$, then RWEM problem (5.9) is convex and strong duality holds.*

Proof. See Appendix B. □

With $\tilde{\mathbf{x}}^*$ and $\mathbf{x}^*$ denoting the optimal solutions for (5.7) and (5.9), we arrive at the following claim.

Proposition 8. *Problem (5.7) is equivalent to (5.9) in the sense that* $\Psi^* = \tilde{\Psi}^*$*, and* $\mathbf{x}^* = \tilde{\mathbf{x}}^*$.

Proof. See Appendix C. □

4.2.2 Lagrange relaxation

Notice that constraints (4.23c)–(4.23e) couple variables across MNs, DCs, workloads, and the RES, so a system operator over the entire network is essential to collect all the information and solve the problem in a centralized way, which may not be feasible in an Internet-scale network [76]. However, since (5.9) is a convex problem [cf. Proposition 7], a Lagrange dual approach can be developed to efficiently find its optimal dual solution with zero duality gap in a decentralized manner [57]. Let $\{\pi_i^t\}$, $\{\lambda_{i,q}^t\}$ and $\{\nu_i^t\}$ denote the Lagrange multipliers associated with the constraints (4.23c)–(4.23e). For notational convenience, let $\lambda_{i,q}^t = 0, \forall i, q \in \mathcal{Q}, t \notin \mathcal{T}_q$. And with $\boldsymbol{\varpi}$ collecting all the Lagrange multipliers, the partial Lagrangian function of (5.9) is

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{\varpi}) := &\sum_{i=1}^{I}\left[G_i(\{R_i^t\}) + \sum_{t=1}^{T} G_{C_i}(P_{i,g}^t) - \sum_{q\in\mathcal{Q}} U_q^t(b_{i,q}^t)\right] \\ &+ \sum_{i=1}^{I}\sum_{t=1}^{T}\nu_i^t\left(d_i^t - \sum_{j=1}^{J} a_{ji}^t - \sum_{q\in\mathcal{Q}} b_{i,q}^t\right) \\ &+ \sum_{i=1}^{I}\sum_{q\in\mathcal{Q}}\sum_{t=1}^{T}\lambda_{i,q}^t\left(\sum_{\tau=S_q}^{t} b_{i,q}^\tau - \sum_{\tau=S_q}^{t}\tilde{b}_{i,q}^\tau\right) \\ &+ \sum_{i=1}^{I}\sum_{t=1}^{T}\pi_i^t\left(P_i^t + P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t - P_{i,g}^t - R_i^t\right).\end{aligned}$$

If $\mathcal{X}$ denotes the set of $\mathbf{x}$ satisfying constraints (4.21b)–(5.7c), and (4.21m)–(4.21o), the Lagrange dual function is given by

$$\mathcal{D}(\boldsymbol{\varpi}) := \min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \boldsymbol{\varpi}) \tag{4.24}$$

and the dual problem of (5.9) is

$$\begin{aligned}
\max\ & \mathcal{D}(\{\pi_i^t\},\{\lambda_{i,q}^t\},\{\nu_i^t\}) \\
\text{s. t. }\ & \pi_i^t \geq 0, \nu_i^t \in \mathbb{R},\ \forall i \in \mathcal{I}, t \in \mathcal{T} \\
& \lambda_{i,q}^t \geq 0,\ \forall i \in \mathcal{I}, q \in \mathcal{Q}, t \in [S_q, E_q - 1] \\
& \lambda_{i,q}^t \in \mathbb{R},\ \forall i \in \mathcal{I}, q \in \mathcal{Q}, t = E_q. \qquad (4.25)
\end{aligned}$$

For the dual problem (5.13), standard subgradient iterations can be employed to obtain the optimal $\boldsymbol{\varpi}^*$, namely

$$\pi_i^t(k+1) = [\pi_i^t(k) + \mu g_{\pi_i^t}(k)]^+,\ \forall i \in \mathcal{I}, t \in \mathcal{T} \qquad (4.26a)$$

$$\lambda_{i,q}^t(k+1) = [\lambda_{i,q}^t(k) + \mu g_{\lambda_{i,q}^t}(k)]^+,\ \forall i, q, t \in [S_q, E_q - 1]$$

$$\lambda_{i,q}^t(k+1) = \lambda_{i,q}^t(k) + \mu g_{\lambda_{i,q}^t}(k), \forall i, q, t = E_q \qquad (4.26b)$$

$$\nu_i^t(k+1) = \nu_i^t(k) + \mu g_{\nu_i^t}(k),\ \forall i \in \mathcal{I}, t \in \mathcal{T} \qquad (4.26c)$$

where k is the iteration index, and $\mu > 0$ is a constant stepsize, and $\{g_{\pi_i^t}(k), g_{\lambda_{i,q}^t}(k), g_{\nu_i^t}(k)\}$ are the subgradients of (5.12) with respect to the Lagrange multipliers. Specifically, we have

$$g_{\pi_i^t}(k) = P_i^t(k) + P_{i,\text{ch}}^t(k) - P_{i,\text{dis}}^t(k) - P_{i,g}^t(k) - R_i^t(k) \qquad (4.27a)$$

$$g_{\lambda_{i,q}^t}(k) = \sum_{\tau=S_q}^{t} b_{i,q}^\tau(k) - \sum_{\tau=S_q}^{t} \tilde{b}_{i,q}^\tau(k) \qquad (4.27b)$$

$$g_{\nu_i^t}(k) = d_i^t(k) - \sum_{j=1}^{J} a_{ji}^t(k) - \sum_{q\in\mathcal{Q}} b_{i,q}^t(k) \qquad (4.27c)$$

where primal variables $\mathbf{x}(k)$ can be obtained as

$$\{a_{ji}^t(k), \tilde{b}_{i,q}^t(k)\}_{i\in\mathcal{I},t\in\mathcal{T},q\in\mathcal{Q}_j} \in \arg \min_{\{a_{ji}^t, \tilde{b}_{i,q}^t\}} \sum_{t=1}^{T}\sum_{i=1}^{I} \left[-a_{ji}^t \nu_i^t(k) - \sum_{q\in\mathcal{Q}_j} \tilde{b}_{i,q}^t \sum_{\tau=t}^{T} \lambda_{i,q}^\tau(k) \right]$$
$$\text{s. t. } \ (4.21f) - (5.7c), (4.21n) - (4.21o) \qquad (4.28)$$

$$\{b_{i,q}^t(k)\}_{t\in\mathcal{T},q\in\mathcal{Q}} \in \arg \min_{\{b_{i,q}^t\}} \sum_{t=1}^{T} \left[b_{i,q}^t \left(\sum_{\tau=t}^{T} \lambda_{i,q}^\tau(k) - \nu_i^t(k) \right) - U_q^t(b_{i,q}^t) \right]$$
$$\text{s. t. } \ (4.21m) \qquad (4.29)$$

and

$$\{R_i^t(k), m_i^t(k), P_{i,\mathrm{ch}}^t(k), P_{i,\mathrm{dis}}^t(k), C_i^t(k), P_{i,g}^t(k), d_i^t(k)\}_{t=1}^T$$
$$\in \arg \min_{\{m_i^t, P_{i,\mathrm{ch}}^t, P_{i,\mathrm{dis}}^t, R_i^t, P_{i,g}^t, d_i^t\}} G_i(\{R_i^t\}) + \sum_{t=1}^T \left[\nu_i^t(k) d_i^t + G_{C_i}(P_{i,g}^t) + \pi_i^t(k)(P_i^t + P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t - P_{i,g}^t - R_i^t)\right]$$
$$\text{s. t.} \quad (4.21b) - (4.21e). \tag{4.30}$$

The subproblems (4.28)–(4.29) are linear programs (LPs) over $\{a_{ji}^t, \tilde{b}_{i,q}^t, b_{i,q}^t\}_{t=1}^T$; hence, they can be optimally solved using available efficient LP solvers. Due to the convexity of $G_i(\{R_i^t\})$, the subproblems (4.30) are convex per DC i. However, since $G_i(\{R_i^t\})$ is non-differentiable due to the absolute value operator and the maximization over $\mathbf{e}_i \in \mathcal{E}_i$, (4.30) still challenges existing solvers. To address this, consider splitting (4.30) into two subproblems as

$$\{m_i^t(k), P_{i,\mathrm{ch}}^t(k), P_{i,\mathrm{dis}}^t(k), C_i^t(k), P_{i,g}^t(k), d_i^t(k)\}_{t=1}^T$$
$$\in \arg \min_{\{m_i^t, P_{i,\mathrm{ch}}^t, P_{i,\mathrm{dis}}^t, P_{i,g}^t, d_i^t\}} \sum_{t=1}^T \left[\pi_i^t(k)(P_i^t + P_{i,\mathrm{ch}}^t - P_{i,\mathrm{dis}}^t - P_{i,g}^t) + \nu_i^t(k) d_i^t + G_{C_i}(P_{i,g}^t)\right]$$
$$\text{s. t.} \quad (4.21b) - (4.21e) \tag{4.31}$$

and

$$\{R_i^t(k)\}_{t=1}^T \in \arg \min_{\{\underline{R}_i \le R_i^t \le \overline{R}_i\}} G_i(\{R_i^t\}) - \sum_{t=1}^T \pi_i^t(k) R_i^t. \tag{4.32}$$

Note that because of the exact relaxation [cf. Proposition 8], $\underline{R}_i$ and $\overline{R}_i$ are lower and upper bounds of the right hand side of (4.23c). Depending on the function $G_{C_i}(P_{i,g}^t)$, subproblem (4.31) is either an LP or a quadratic program. Hence, the optimal solution can be obtained by existing solvers. And for nonsmooth subproblems (4.32), a standard subgradient iteration can be employed to obtain the optimal solution as

$$R_i^t(\ell + 1) = R_i^t(\ell) - \mu(\ell) g_{R_i^t}(\ell), \ \forall t \in \mathcal{T} \tag{4.33}$$

where ℓ denotes iteration index, and $\{\mu(\ell)\}$ is a non-summable but square-summable stepsize se-

Algorithm 4 Subgradient iteration for solving (4.32)

1: **Initialize:** Generate all the vertices of the polyhedral uncertainty set $\mathcal{E}_i$; choose a proper $\{R_i^t(0)\}$ and stepsize sequence $\mu(\ell)$

2: **repeat** $\ell = 0, 1, 2 \ldots$

3: Evaluate all the vertices in $\mathcal{E}_i$ and find $\mathbf{e}_i^*(\ell)$ in (4.35)

4: Calculate subgradients via (4.34)

5: Update $\{R_i^t(\ell)\}$ via (4.33)

6: **until** Convergence

quence; while the partial subgradient of $G_i(\{R_i^t\})$ with respect to R_i^t is obtained as

$$\begin{aligned} g_{R_i^t}(\ell) :=& \frac{\partial \left(G_i(\{R_i^t\}) - \sum_{t=1}^{T} \pi_i^t(k) R_i^t \right)}{\partial R_i^t} \\ =& \begin{cases} \alpha_i^t - \pi_i^t(k), \text{if } R_i^t(\ell) \geq E_i^{t*}(\ell) \\ \beta_i^t - \pi_i^t(k), \text{if } R_i^t(\ell) < E_i^{t*}(\ell) \end{cases} \end{aligned} \tag{4.34}$$

where $\mathbf{e}_i^*(\ell) := [E_i^{1*}(\ell), \ldots, E_i^{T*}(\ell)]'$ for the given $\{R_i^t(\ell)\}$ is found using

$$\mathbf{e}_i^* \in \arg \max_{\mathbf{e}_i \in \mathcal{E}_i} \sum_{t=1}^{T} \left(\psi_i^t | R_i^t(\ell) - E_i^t | + \phi_i^t (R_i^t(\ell) - E_i^t) \right). \tag{4.35}$$

It can be seen that the objective function in (4.35) is convex in $\mathbf{e}_i$ under the condition $\alpha_i^t \geq \beta_i^t$, $\forall t \in \mathcal{T}$. However, computing where the maximum of a convex function is attained can be *NP-hard*, in general. Fortunately, the globally optimal solution is attainable at the extreme points of $\mathcal{E}_i$ for convex maximization [15, Sec. 2.4]. Leveraging the polyhedral structure of $\mathcal{E}_i$, we utilize an efficient vertex enumerating algorithm to evaluate the objective in (4.35), and obtain $\mathbf{e}_i^*$ directly; see Algorithm 7. Although the number of vertices may increase exponentially with the number of variables and constraints, all the vertices of $\mathcal{E}_i$ need be generated only once before running Algorithm 7, which means that Algorithm 7 is computational affordable. In fact, our simulations in Section 4.3 will corroborate that the vertex generating procedure can be completed within several seconds.

Algorithm 5 Distributed workload and energy management

1: **Initialize:** Choose a proper $\boldsymbol{\varpi}(0)$ and stepsize μ

2: **repeat** $k = 0, 1, 2 \ldots$

3: Each DC obtains $\{b_{i,q}^t(k), R_i^t(k), m_i^t(k), P_{i,\text{ch}}^t(k), P_{i,\text{dis}}^t(k), C_i^t(k), P_{i,g}^t(k), d_i^t(k)\}$ by solving (4.29) and (4.31)-(4.32) separately

4: Each MN solves (4.28) and sends $\{a_{ji}^t(k), \tilde{b}_{i,q}^t(k)\}$ to each DC

5: DCs update $\boldsymbol{\varpi}(k)$ via (4.26) and send them to MNs

6: Run averages to recover primal variables via (4.36)

7: **until** Convergence

4.2.3 Optimality and distributed implementation

For the subgradient iterations (4.33), if a diminishing stepsize satisfying (i) $\sum_{\ell=0}^{\infty} \mu(\ell) = \infty$, and (ii) $\sum_{\ell=0}^{\infty} \mu(\ell)^2 < \infty$ is adopted, the sequence (4.33) converges as $\ell \to \infty$ to the optimal $\{R_i^t(k)^*\}$ [15]. As a constant stepsize μ is used in (4.26), the subgradient iterations will converge to a neighborhood of the optimal solution $\boldsymbol{\varpi}^*$ [15]. The size of the neighborhood is proportional to the stepsize μ. Since the objective (4.23a) is not strictly convex, running averages of the primal sequence $\{\mathbf{x}(k)\}$ can be used to recover the optimal primal solutions, which are given by

$$\overline{\mathbf{x}}(k) = \frac{1}{k}\mathbf{x}(k-1) + \frac{k-1}{k}\overline{\mathbf{x}}(k-1), \ \forall k \tag{4.36}$$

where $\overline{\mathbf{x}}(k)$ is the average of all primal solutions up to iteration $k-1$. Since set $\mathcal{X}$ is convex and $\mathbf{x}(i) \in \mathcal{X}, \ 0 \leq i \leq k-1$, it turns out that $\overline{\mathbf{x}}(k)$ is a feasible point in $\mathcal{X}$. In addition, it can be shown that $\overline{\mathbf{x}}(k)$ is also asymptotically feasible for primal problem (5.9) [53].

It is also worth noting that RWEM can afford a distributed implementation, where optimization tasks are distributed among MNs and individual DCs; see Algorithm 6. In RWEM, dual variable updates (4.26) are all implemented at each DC locally. Subproblem (4.28) is solved by each MN operator, while each DC operator solves subproblems (4.29) and (4.31)–(4.32). To make these distributed implementations possible, a bidirectional message passing between MNs and DCs is necessary. At every iteration, workload routing variables $\{a_{ji}^t(k), \tilde{b}_{i,q}^t(k)\}$ are sent from each node to each DC, while the dual variables $\{\lambda_{i,q}^t\}$ and $\{\nu_i^t\}$ are fed back to each MN in turn to solve (4.28). Note that instead of real-time power and workload schedules, only Lagrange multipliers are

Table 4.1: Power supply parameters. The units are kW.

DC i	$\overline{P}_{i,g}$	$\overline{C}_i$	$\underline{C}_i$	$\overline{P}_{i,\mathrm{ch}}$	$\overline{P}_{i,\mathrm{dis}}$	η	δ	R_i^{up}	R_i^{dw}
1	30	5	40	20	20	0.95	0.95	24	24
2	30	5	40	20	20	0.95	0.95	24	24
3	30	5	40	20	20	0.95	0.95	24	24
4	30	5	40	20	20	0.95	0.95	24	24

Table 4.2: Energy purchase prices. The units are $/kWh.

Slot	1	2	3	4	5	6	7	8	9	10	11	12
α_1^t	0.53	0.50	0.50	0.53	0.65	0.90	0.82	0.73	0.68	0.59	0.51	0.45
α_2^t	0.49	0.46	0.43	0.43	0.45	0.56	0.77	0.70	0.63	0.58	0.51	0.44
α_3^t	0.43	0.42	0.39	0.36	0.37	0.37	0.46	0.65	0.58	0.53	0.48	0.43
α_4^t	0.69	0.68	0.65	0.61	0.58	0.57	0.60	0.75	1.02	0.94	0.84	0.78

sent back to MNs, which could be further leveraged to enhance privacy-preserving operations.

In addition, the worst-case complexity of solving a general convex program is on the order $\mathcal{O}\left(\max\{N_c, N_v\}^4\sqrt{N_v}\log(1/\epsilon)\right)$, where N_v, N_c are the total number of variables and constraints, and $\epsilon > 0$ is the given accuracy [74]. Hence, solving subproblems (4.28)–(4.29), (4.31)–(4.32) in a distributed fashion incurs a markedly lower complexity than directly tackling (5.9) in a centralized fashion. Faster implementations are possible if we further decentralize (4.31), and let operators of conventional generation, storage units, delay-tolerant workloads, solve subproblems separately.

4.3 Numerical Evaluation

In this section, results of simulated tests are presented to demonstrate the merits of the proposed approach.

Table 4.3: Delay-tolerant workloads parameters.

	DW 1	DW 2	DW 3	DW 4	DW 5	DW 6	DW 7	DW 8
Node	1	1	2	2	3	3	4	4
B_j	75	100	87.5	112.5	100	75	75	150
$[S_q, E_q]$	2-9	1-4	4-10	4-6	1-12AM	2-6	2-7	1-5
$\overline{u}_q, \underline{u}_q$	0.95,0.6	1.1,0.8	0.67,0.61	0.68,0.58	0.6,0.05	0.5,0.5	0.4,0.4	0.3,0.3

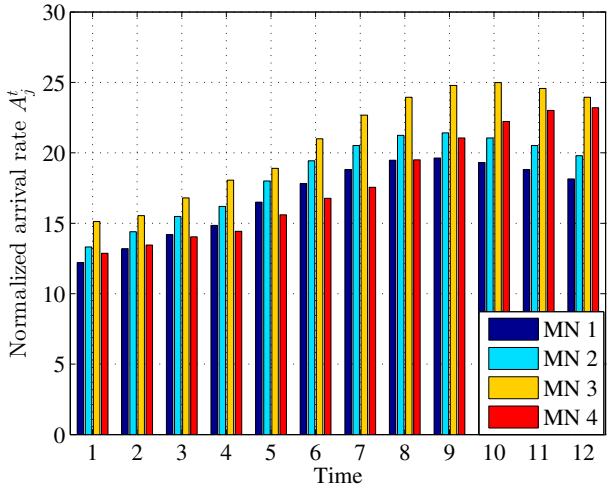

Figure 4.3: Real-time arrival rate of interactive workloads.

4.3.1 Experiment setup

The Matlab-based modeling package `CVX 2.1` [5] is used to solve the optimization problems involved. The DC network includes 4 DCs and 4 MNs uniformly located in the eastern, central, mountain and western parts of the US. Each DC is connected to a microgrid, of which the power supply parameters are listed in Table 4.1. A polyhedral uncertainty set (4.13) with a single sub-horizon (no partition, $\mathcal{T}_{i,s} = \mathcal{T}$) is considered for the RES. The upper/lower limits $\{\overline{E}_i^t, \underline{E}_i^t, \overline{E}_{\mathcal{T}_{i,s}}, \underline{E}_{\mathcal{T}_{i,s}}\}$ and the average RES over scheduling horizon $E_{\mathcal{T}_{i,s}}^{\text{avg}}$ were rescaled from the CAISO solar generations during Mar. 1-Oct. 30, 2012; see [6] for detailed description. The upper and lower levels of robustness are set by default to $\Delta_s^{\text{up}} = \Delta_s^{\text{low}} = 1$. Table 4.2 lists the energy purchase prices α_i^t, which are obtained by scaling the hourly electricity prices of the New York City [7]. The selling price is set to $\beta_i^t = \xi \alpha_i^t$ with $\xi = 0.6$, while the CG cost is considered as $G_{c_i}(P_{i,g}^t) = \omega_i P_{i,g}^t$ with $\omega_i = (1/T)\sum_{t=1}^T \alpha_i^t$.

The total number of servers $\overline{M}_i$ is set to 80, with common $\varrho = 0.4$, $\overline{P}_{i,s} = 500\text{W}$ and $\underline{M}_i = 5$

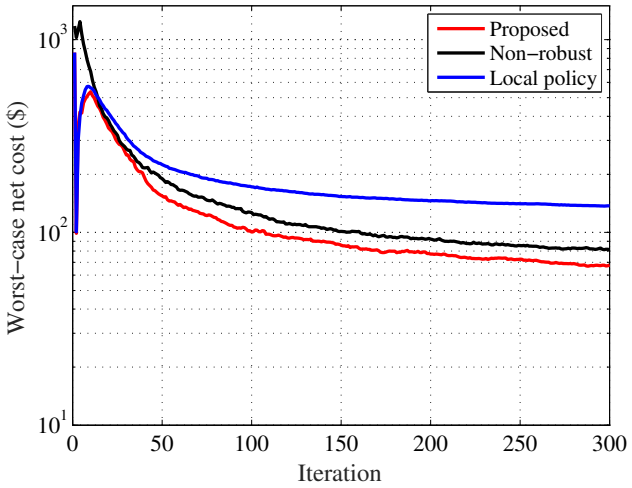

Figure 4.4: Comparison of worst-case net costs.

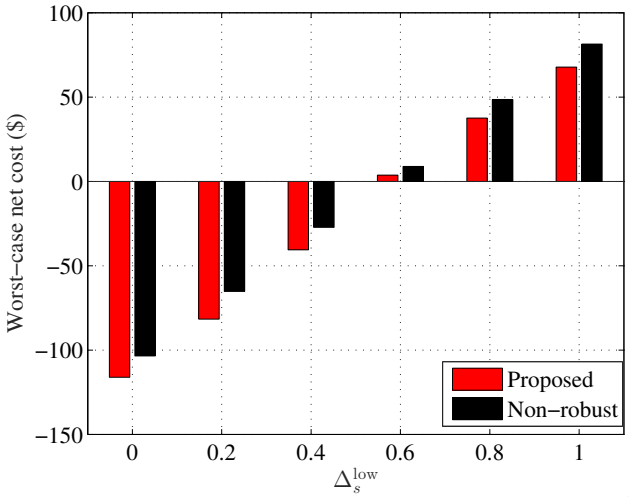

Figure 4.5: Worst-case net costs versus the level of robustness Δ_s^{low}.

for all DC i. The cooling parameters are set to $\gamma = 0.2, \kappa_i^t = 2 \times 10^{-9}, \overline{P}_{i,a}^t = 30\text{kW}, \forall i \in \mathcal{I}, t \in \mathcal{T}$. For simplicity, the unit of workloads in this setting is normalized by the server is per-slot capacity D_i, with common D_i for all DCs. The interactive workload arrival rates at MNs are depicted in Fig. 4.3, rescaled from the real traffic of Wikipedia [82]. Eight different delay-tolerant workloads (DWs) are specified in Table 4.3. For job q, the revenue function is considered $U_q^t(b_{i,q}^t) = -0.01u_q^t(b_{i,q}^t)^2 + u_q^t b_{i,q}^t$ \$/unit, where u_q^t is linearly decreasing from $\overline{u}_q$ to $\underline{u}_q$ across its

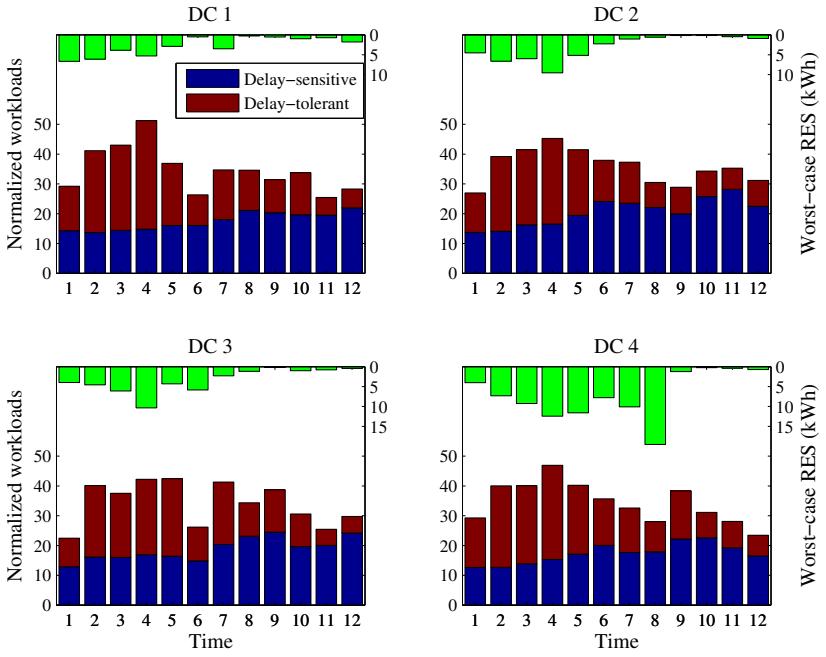

Figure 4.6: Optimal workload schedule d_i^t of the proposed algorithm.

active interval $[S_q, E_q]$. The bandwidth limits L_{ji}^t are assumed to be time-invariant, given by

$$\mathbf{L}^t = \begin{bmatrix} 90 & 60 & 40 & 40 \\ 40 & 80 & 40 & 50 \\ 50 & 40 & 100 & 30 \\ 40 & 40 & 50 & 90 \end{bmatrix}, \ \forall t \in \mathcal{T}. \tag{4.37}$$

Note that the homogenous settings of data centers are considered here in order to exemplify the impact of other factors (e.g., prices and RES).

The time horizon spans $T = 12$ hours, corresponding to the interval 1PM–12AM in Eastern Time Zone. Here we use the Eastern Time Zone for time-keeping, and the real data have been shifted to show the effect of time zone differences. As a result, the peaks of workload demands, RES and prices are different in the four areas, which provide an opportunity for spatio-temporal workload and energy management. Finally, two benchmarks are compared in this setting: A robust *local policy* allocating all workloads from each MN to its nearest DC, and a *non-robust* geographical load balancing policy, which predicts the RES at each slot via its sample mean from the historical dataset. Interestingly, our proposed approach can be reduced to the local policy if the bandwidth matrix $\mathbf{L}^t$

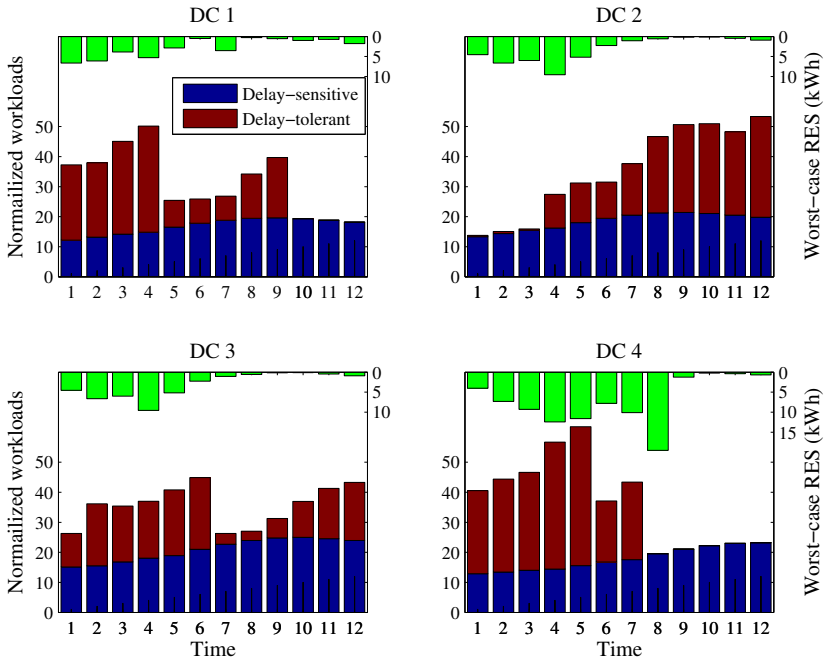

Figure 4.7: Optimal workload schedule d_i^t of the local policy.

only has positive diagonal entries, and to the non-robust policy if we set $\Delta_s^{\text{up}} = 0, \Delta_s^{\text{low}} = 0$, and $|\mathcal{T}_{i,s}| = 1$ in (4.13).

4.3.2 Numerical tests

Fig. 4.4 depicts the evolution of the worst-case net cost for the proposed algorithm, as well as the two alternatives. Within 300 iterations, the proposed algorithm converges to a worst-case net cost 19% lower than that of the non-robust approach, and 51% lower than that of the local policy. Recall that the non-robust approach is sensitive to the RES prediction error, while the local policy can not perform geographical load balancing. In contrast, the proposed RWEM takes advantage of both factors, and purchases less amount of expensive energy from the spot market that results in a smaller worst-case net cost.

To better illustrate this point, sensitivity analysis to the level of robustness Δ_s^{low} is first studied in Fig. 4.5. As expected, the proposed RWEM outperforms the non-robust approach in all cases. Meanwhile, the worst-case net costs of both robust and non-robust scheme grow up as the Δ_s^{low} increases. This makes sense intuitively because a larger Δ_s^{low} implies a bigger uncertainty set [cf.

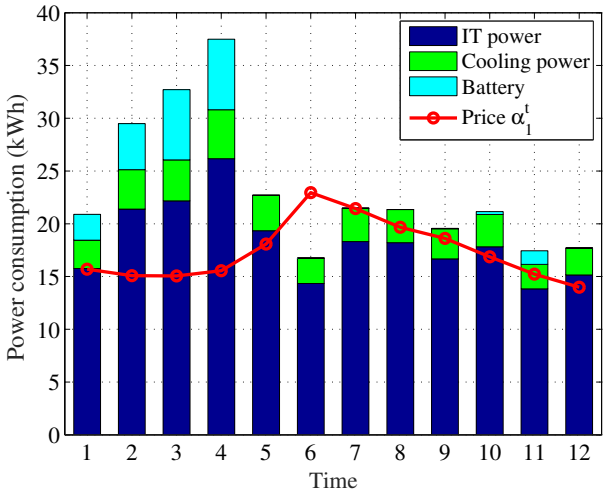

Figure 4.8: Optimal power consumption schedule in DC 1.

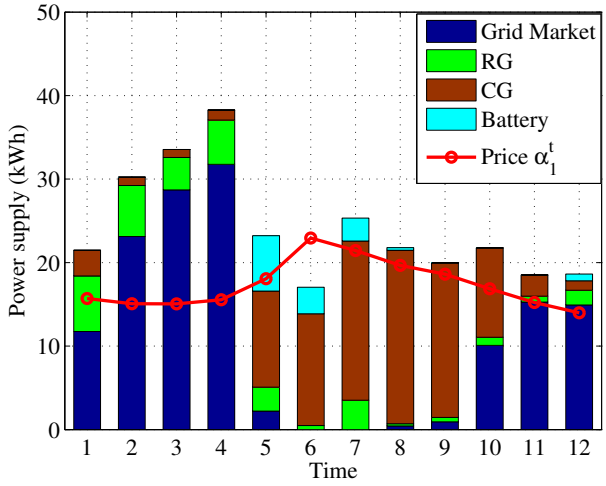

Figure 4.9: Optimal power supply schedule in DC 1.

(4.13)], which will eventually increase the worst-case net cost. Hence, the selection of Δ_s^{low} is critical for various scenarios. While a large Δ_s^{low} guarantees robustness of the resultant solution, a small one can moderately reduce its conservatism. We waive the analysis of Δ_s^{up} here, because it plays a less important role in cost minimization (5.7), due to the monotonicity of the objective (5.7a) with respect to E_i^t.

The optimal workload schedules d_i^t of the proposed RWEM and the local policy are compared in Figs. 4.6 and 4.7, respectively. The worst-case renewable generations $\{E_i^{t*}\}$ are shown to illustrate the principle of geographical workload distribution. Compared with the local policy, the proposed

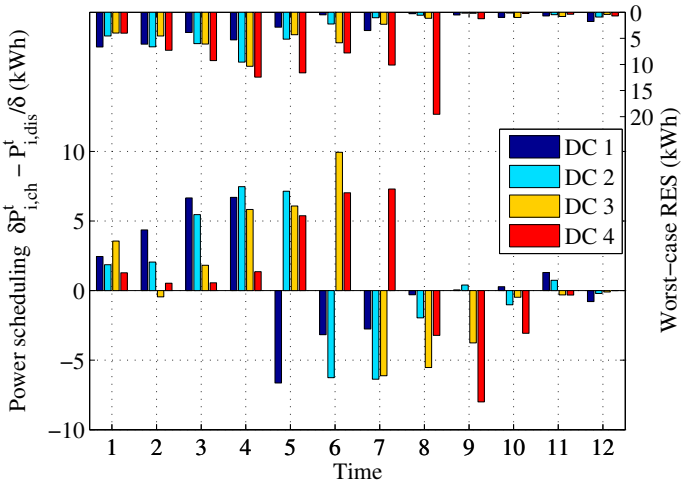

Figure 4.10: Optimal battery (dis-)charging schedule in all DCs.

RWEM can intelligently route workloads to a remote DC where the system IT demand is lower, RES availability is higher, or, the local energy price is more affordable. To see this, both interactive and flexible workloads are uniformly routed to each DC in Fig. 4.6, thus the entire DC network can process more workloads when the RES generations are ample (1PM-6PM). An interesting observation is that even if DC 4 in the Western US enjoys a relatively higher RES, due to its high energy purchase price α_i^t [cf. Table 4.2], the MNs are more likely to route workloads to the areas having lower prices, when there is no renewable surplus in DC 4.

In contrast to Fig. 4.6, the workload schedules in Fig. 4.7 are more isolated and thus inefficient. Without coordinating all DCs, the local policy cannot "smooth" the IT demand with the additional degree of freedom in space. Specifically, when the system demand is low in DC 1 as well as DC 3 (10-12 PM), no flexible workloads can be scheduled. Likewise, the RES is not fully utilized in DC 2 (1-3PM). This high fluctuation of system workloads will also cause switching on/off servers frequently, thus incurring an implicitly higher wear-and-tear cost.

The optimal power consumption and supply schedules of DC 1 are depicted in Figs. 4.8 and 4.9, respectively. The scaled fluctuation of energy purchase price α_1^t is also plotted to gain intuition on the optimal power schedules. Clearly, less power is consumed when α_1^t is higher (6PM). Using combined cooling sources, the average cooling coefficient of the proposed algorithm is around 0.17, which is more efficient than the simple chilled-water cooling with a constant coefficient $\gamma = 0.2$.

Furthermore, with the goal of mitigating the high variability of RES, batteries are encouraged to charge when the worst-case renewable generations are high and the energy prices are low (1PM-4PM). Thus, batteries can be discharged when less renewables are available at night (e.g., 5-7PM in Fig. 4.9). Likewise, Fig. 4.10 shows that all the batteries exhibit a similar trend in response to RES and price fluctuations. From the power supply perspective, the lower purchase price α_1^t encourages purchasing more energy from the external grid market, while the peak of α_1^t results in a higher power usage from the CG.

4.4 Appendices

4.4.1 Proof of Lemma 10

Function $P_{i,IT}(d_i^t, m_i^t)$ is jointly convex in $\{d_i^t, m_i^t\}$ because of its quadratic-over-linear form [cf. (4.7)]. Hence, the composite function $\mathcal{F}_i^t(P_{i,IT}(d_i^t, m_i^t))$ is jointly convex in $\{d_i^t, m_i^t\}$ since $\mathcal{F}_i^t(P_{i,IT}^t)$ is convex and nondecreasing [19, Sec. 3.2]; and, so is $P_i^t(d_i^t, m_i^t)$.

4.4.2 Proof of Proposition 1

Since the absolute value function is convex, and the operations of nonnegative weighted summation and pointwise maximum preserve convexity, it is easy to see that $G_i(\{R_i^t\})$ is convex in $\{R_i^t\}$. In addition, $G_{C_i}(P_{i,g}^t)$ is convex in $P_{i,g}^t$ and $U_q^t(b_{i,q}^t)$ is concave in $b_{i,q}^t$; hence, the objective function (4.23a) is jointly convex in $\{R_i^t, P_{i,g}^t, P_{i,\text{ch}}^t, P_{i,\text{dis}}^t\}$. The constraints except $(4.23c)$ are linear, while $(4.23c)$ defines a feasible set which is actually the epigraph of a convex function [cf. Lemma 10]. Since the epigraph of a convex function is a convex set [19, Chap. 3.1], it follows that (5.9) is a convex problem, and strong duality holds.

4.4.3 Proof of Proposition 2

Compared to (5.7), (5.7d)–(4.21k) are replaced by (4.23d), and (4.21i)–(4.21j) are substituted by (4.23c) to convexify the problem, and facilitate distributed implementation in (5.9). By summing (5.7d) from S_q to $t \in [S_q, \ldots, E_q - 1]$, we find $Q_{i,q}^t = Q_{i,q}^{S_q} - \sum_{\tau=S_q}^t b_{i,q}^\tau + \sum_{\tau=S_q}^t \tilde{b}_{i,q}^\tau$. Then, due to (4.21k), we have that $\sum_{\tau=S_q}^t \tilde{b}_{i,q}^\tau \geq \sum_{\tau=S_q}^t b_{i,q}^\tau$. Likewise, we obtain $\sum_{\tau=S_q}^{E_q} \tilde{b}_{i,q}^\tau = \sum_{\tau=S_q}^{E_q} b_{i,q}^\tau$,

which establishes the equivalence of (5.7d)-(4.21k) with (4.23d). In addition, since the objective (4.23a) is monotonically increasing with R_i^t, it is easy to see that (4.23c) is always binding at the optimal solution $\mathbf{x}^*$, which implies that the optimal solution $\mathbf{x}^*$ is also an optimal solution (5.7); and thus, $\Psi^* = \tilde{\Psi}^*$.

Chapter 5

Learning-aided Stochastic Optimization over Data Center Networks

5.1 Modeling Preliminaries

Our proposed approach is applicable to more general network resource allocation such as energy management in power networks, cross-layer resource allocation in communication networks, and traffic control in a transportation networks. For the purpose of motivating, we start by introducing a specific resource allocation in current sustainable DC networks. Consider a system operating on discrete time slots indexed by t, with an infinite scheduling horizon $\mathcal{T} := \{0, 1, \ldots\}$. A network with $\mathcal{J} := \{1, 2, \cdots, J\}$ mapping nodes (MNs), and $\mathcal{I} := \{1, 2, \cdots, I\}$ heterogeneous data centers (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).

5.1.1 Traffic workloads and network constraints

Suppose that MNs collect both interactive workloads and delay-tolerant workloads [85]. Interactive workloads are real-time requests that must be served immediately. Delay-tolerant workloads are relatively time insensitive and deferrable within given slots. Typical examples include system updates and data backup, which provide ample optimization opportunities for workload allocation based

on the dynamic variation of energy prices and renewable energy availability. In this chapter, we mainly focus on the scheduling of delay-tolerant workloads, and assume that interactive workloads are perfectly performed via the existing approaches [82].

In this case, let $v_{j,t}$ denotes the amount of delay-tolerant workload arriving at MN j at slot t, and the vector $\tilde{\mathbf{x}}_{j,t} := [\tilde{x}_{1,j,t}, \ldots, \tilde{x}_{I,j,t}]^\top \in \mathbb{R}^I$ denotes the amount routed from MN j to each DC at slot t, respectively. The delay-tolerant workloads might be different and the fraction of unserved workload is buffered in queues (one per class of delay-tolerant workload). The dynamic recursion model for the queues can be formalized as

$$Q^{\mathrm{mn}}_{j,t+1} = \left[Q^{\mathrm{mn}}_{j,t} + v_{j,t} - \textstyle\sum_{i \in \mathcal{I}_j} \tilde{x}_{i,j,t} \right]^+, \ \forall j, t, \tag{5.1}$$

where $Q^{\mathrm{mn}}_{j,t}$ is the queue length of delay-tolerant workload in MN j at the beginning of slot t.

At the DC side, with $x_{i,t}$ denoting the amount of delay-tolerant workload 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

$$Q^{\mathrm{dc}}_{i,t+1} = \left[Q^{\mathrm{dc}}_{i,t} - x_{i,t} + \textstyle\sum_{j \in \mathcal{J}} \tilde{x}_{i,j,t} \right]^+, \ \forall i, t, \tag{5.2}$$

where $Q^{\mathrm{dc}}_{i,t}$ is the queue length of delay-tolerant workload in DC i at the beginning of slot t. And the instantaneous process rate is bounded by the capacity of each DC through

$$0 \leq x_{i,t} \leq D_i, \ \forall i, t \tag{5.3}$$

where the D_i is the computational capacity of DC i.

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 $B_{i,j}$, i.e.,

$$0 \leq \tilde{x}_{i,j,t} \leq B_{i,j}, \quad \forall j, i, t. \tag{5.4}$$

Note that we set $B_{i,j} = 0$ if MN j and DC i are not connected.

5.1.2 Operational costs

Starting with the service and distribution of the workloads, distribution of workloads across the network generates bandwidth *costs*. To this end, we will use the convex function $G^{\mathrm{d}}_{i,j}(\,\cdot\,)$ to denote

the cost for distributing workloads from MN j to DC i, which depends on the distance between them.

Let $P_{i,t}^{\mathrm{r}}$ denote the renewable energy generated at the beginning of slot t by the renewable generator in DC i, which is upper-bounded by $P_{i,t}^{\mathrm{r}} \leq \overline{P}_i^{\mathrm{r}}$, $\forall t$. For the server power consumption in DC i, we approximate it as a quadratic function of the total IT demand $P_i^{\mathrm{dc}}(x_{i,t}) = e_{i,t}x_{i,t}^2$, where $e_{i,t}$ is time-varying parameter capturing a variety of environment factors (e.g., humidity, temperature) [21]. The *energy transaction cost* is modeled as a linear function of the power imbalance amount $|P_i^{\mathrm{dc}}(x_{i,t}) - P_{i,t}^{\mathrm{r}}|$ to capture the cost in real time power balancing

$$G_{i,t}^{\mathrm{e}}(x_{i,t}) := \alpha_{i,t}\left(e_{i,t}x_{i,t}^2 - P_{i,t}^{\mathrm{r}}\right) \tag{5.5}$$

where $\alpha_{i,t}$ is the buy/sell price in the local power wholesale market. Clearly, each DC should buy energy from external energy markets in slot t at price $\alpha_{i,t}$ if $P_i^{\mathrm{dc}}(x_{i,t}) - P_{i,t}^{\mathrm{r}} > 0$, or, sell energy to the markets with the same price if $P_i^{\mathrm{dc}}(x_{i,t}) - P_{i,t}^{\mathrm{r}} < 0$.

The resultant aggregated *network cost* for the considered MN-DC network *at time* t is a convex function of $\mathbf{x}_t$, namely

$$\Psi_t(\{x_{i,t}\}, \{\tilde{\mathbf{x}}_{j,t}\}) := \sum_{i\in\mathcal{I}} G_{i,t}^{\mathrm{e}}(x_{i,t}) + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} G_{i,j}^{\mathrm{d}}\left(\tilde{\mathbf{x}}_{j,t}\right). \tag{5.6}$$

5.2 Stochastic Resource Allocation over Networks

Building upon the mathematic network models in Section 5.1, 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 $\{\tilde{\mathbf{x}}_{j,t}\}$ and workloads scheduling $\{x_{i,t}\}$. The goal is to minimize the limiting average network cost, subject to IT operational constraints.

To streamline, we collect all sources of randomness into the state vector $\boldsymbol{s}_t := \{e_{i,t}, P_{i,t}^{\mathrm{r}}, v_{j,t}, \forall i,j\} \in \mathbb{R}^{2I+J}$, and also define the concatenated optimization variables $\mathbf{x}_t := [\tilde{\mathbf{x}}_{1,t}^\top, \ldots, \tilde{\mathbf{x}}_{J,t}^\top, x_{1,t}, \ldots, x_{I,t}]^\top \in \mathbb{R}^{IJ+I}$ and $\mathbf{Q}_t := \{Q_{j,t}^{\mathrm{mn}}, Q_{i,t}^{\mathrm{dc}},\ \forall i,j\} \in \mathbb{R}^{I+J}$. Considering the scheduled workloads $\{x_{i,t}\}$ as the amount of outgoing link at each DC i, we can define a "node-incidence" matrix $\mathbf{A} \in \mathbb{R}^{(I+J)\times(IJ+I)}$. In particular, the components of the matrix $\mathbf{A}$ are such that the first J rows for each MN and the last

I rows for each DC and the first $I \times J$ columns for the links between MN j to DC i and the last I columns for the outgoing link of each DC i, namely

$$\mathbf{A}_{(i,e)} = \begin{cases} 1, \text{ if link } e \text{ enters node } i \\ -1, \text{ if link } e \text{ leaves node } i \\ 0, \text{ else} \end{cases}$$

where $\mathbf{A}_{(i,e)}$ is the i-th row and e-th column entry of the matrix $\mathbf{A}$. Also, the instantaneous workload arrival rates are collected in the vector $\mathbf{c}_t := [v_{1,t}, \ldots, v_{J,t}, 0, \ldots, 0]^\top$, while the capacities of workload distribution and processing are stacked in $\bar{\mathbf{x}} := [B_{1,1}, \ldots, B_{i,j}, D_1, \ldots, D_I]^\top$. Hence, the optimal scheduling is obtained as the solution of the following *long-term* network-optimization problem

$$\Psi^* := \min_{\{\mathbf{x}_t, \mathbf{Q}_t, \forall t\}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right] \tag{5.7a}$$

$$\text{s. t.} \quad 0 \leq \mathbf{x}_t \leq \bar{\mathbf{x}}, \ \forall t \tag{5.7b}$$

$$\mathbf{Q}_{t+1} = [\mathbf{Q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{c}_t]^+, \ \forall t \tag{5.7c}$$

$$\lim_{T\to\infty} \frac{1}{T} \textstyle\sum_{t=0}^{T-1} \mathbb{E}\left[\mathbf{Q}_t\right] < \infty \tag{5.7d}$$

The objective in (5.7a) 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 $\boldsymbol{s}_t$, and the possible randomness of the control policy). Although strictly speaking the problem in (5.7) is convex, the delay-tolerant workload queues in (5.7c) couple the optimization variables over the infinite time horizon. Even worse, for the practical case where the knowledge of $\mathbf{s}_t$ is causal, finding the optimal solution requires using dynamic programming tools, which are generally intractable. Our approach to circumventing this obstacle is to relax (5.7c) by replacing them with the average constraints and employ dual decomposition techniques to separate the solution across time. This is elaborated in the next section.

5.2.1 Problem reformulation

Combining (5.7c) and (5.7d), it follows that in the long term time average workload arrival and departure rates must satisfy the following necessary condition

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\mathbf{A}\mathbf{x}_t+\mathbf{c}_t\right]\leq 0. \tag{5.8}$$

In other words, in the long term all buffered delay-tolerant workloads should be served. Using (5.8), we can write the relaxed version of (5.7) as

$$\tilde{\Psi}^*:=\min_{\{\mathbf{x}_t\}}\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right]\quad \text{s.t. } (5.7b),(5.8). \tag{5.9}$$

Compared to (5.7), the state variables $\mathbf{Q}_t$ are not present in (5.9), and the time-coupling constraints (5.7c) are replaced with (5.8). The problem in (5.9) has a number of interesting properties, including: a) as (5.9) is a relaxed version of (5.7), it follows that $\tilde{\Psi}^*\leq\Psi^*$; and b) if $\{\boldsymbol{s}_t\}$ is stationary, the solution is also stationary and easy to characterize. Regarding property b), using arguments similar to those in, e.g. [54, 72], it can be shown that if the random process $\boldsymbol{s}_t$ is stationary, there exists a stationary control policy $\boldsymbol{\chi}(\cdot)$ which is a pure function of the current $\boldsymbol{s}_t$; satisfies (5.7b); and guarantees that $\mathbb{E}[\Psi_t(\boldsymbol{\chi}(\boldsymbol{s}_t))]=\tilde{\Psi}^*$, and $\mathbb{E}[\mathbf{A}\boldsymbol{\chi}(\boldsymbol{s}_t)+\mathbf{c}_t(\mathbf{s}_t)]\leq 0$. This implies that the *dynamic* stochastic problem (5.9) is equivalent to the following *static* convex stochastic program

$$\tilde{\Psi}^*:=\min_{\boldsymbol{\chi}(\cdot)}\ \mathbb{E}\left[\Psi_t(\boldsymbol{\chi}(\boldsymbol{s}_t))\right] \tag{5.10a}$$

$$\text{s.t.}\quad \mathbb{E}[\mathbf{A}\boldsymbol{\chi}(\boldsymbol{s}_t)+\mathbf{c}_t(\mathbf{s}_t)]\leq 0 \tag{5.10b}$$

$$0\leq\boldsymbol{\chi}(\boldsymbol{s}_t)\leq\bar{\mathbf{x}},\ \forall\mathbf{s}_t \tag{5.10c}$$

where we interchangeably use $\boldsymbol{\chi}(\boldsymbol{s}_t)=\mathbf{x}_t$ to emphasize the dependence of the real-time decision $\mathbf{x}_t$ on the random state $\boldsymbol{s}_t$. Note that the optimization in (5.10) is with respect to the stationary policy (or the probability distribution) $\boldsymbol{\chi}(\cdot)$. Hence, there is an infinite number of variables in the primal domain. Observe though, that there is a finite number of constraints coupling the realizations [cf. (5.10b)]. Thus, the dual problem contains a finite number of variables hinting that the problem is likely more tractable in the dual space [30].

5.2.2 Lagrangian dual and optimal solutions

Let $\boldsymbol{\lambda} := [\lambda_1^{\mathrm{mn}}, \ldots, \lambda_J^{\mathrm{mn}}, \lambda_1^{\mathrm{dc}}, \ldots, \lambda_I^{\mathrm{dc}}]^\top$ denote the Lagrange multipliers associated with constraints (5.10b). If we define $\mathbf{x} := \{\mathbf{x}_t, \forall t\}$, the partial Lagrangian function of (5.9) is $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) := \mathbb{E}\Big[\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})\Big]$. Hence, the instantaneous Lagrangian can be defined as

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) := \Psi_t(\mathbf{x}_t) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t). \tag{5.11}$$

Considering $\mathcal{X}$ as the feasible set defined by the instantaneous constraints in (5.7b) or (5.10c), which are the ones not dualized in (5.11), the dual function $\mathcal{D}(\boldsymbol{\lambda})$ can be written as

$$\mathcal{D}(\boldsymbol{\lambda}) := \min_{\{\mathbf{x}\in\mathcal{X}\}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) := \mathbb{E}\left[\min_{\mathbf{x}_t\in\mathcal{X}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})\right]. \tag{5.12}$$

Likewise, the dual problem of (5.9) is

$$\max_{\boldsymbol{\lambda}\geq 0} \mathcal{D}(\boldsymbol{\lambda}) := \mathbb{E}\left[\mathcal{D}_t(\boldsymbol{\lambda})\right] \tag{5.13}$$

where $\mathcal{D}_t(\boldsymbol{\lambda}) := \min_{\mathbf{x}_t\in\mathcal{X}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$.

To this end, if the optimal Lagrange multipliers $\boldsymbol{\lambda}^*$ is known, then a sufficient condition for the optimal solution of (5.9) or (5.10) is to minimize the Lagrangian function $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*)$ or its instantaneous version $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}^*)$ over the set $\mathcal{X}$ [16, Proposition 3.3.4]. Specifically, the optimal routing $\{\tilde{\mathbf{x}}_{j,t}^*\}$ and workloads scheduling $\{x_{i,t}^*\}$ in the considered DC networks can be characterized as a function of the optimal multipliers $\boldsymbol{\lambda}^*$ associated with the constraints (5.10b), and the realization of the random state $\boldsymbol{s}_t$, which is explicitly described in the following proposition.

Proposition 9. *Consider the optimization problem in (5.10). Given the instantaneous realization for the random state* $\mathbf{s}_t$*, and the Lagrange multipliers* $\boldsymbol{\lambda}^*$ *associated with the constraints (5.10b), the optimal instantaneous workload routing decisions are given by*

$$\tilde{x}_{i,j,t}^*(\mathbf{s}_t) = \left[(\nabla G_{i,j}^{\mathrm{d}})^{-1}\Big((\lambda_j^{\mathrm{mn}})^* - (\lambda_i^{\mathrm{dc}})^*\Big)\right]_0^{B_{i,j}} \tag{5.14}$$

and the optimal instantaneous workload scheduling decisions are given by

$$x_{i,t}^*(\mathbf{s}_t) = \left[\Big(\nabla G_i^{\mathrm{e}}(\mathbf{s}_t)\Big)^{-1}\Big((\lambda_i^{\mathrm{dc}})^*\Big)\right]_0^{D_i}. \tag{5.15}$$

We omit the proof of Proposition 9, which can be easily derived using KKT conditions for constrained optimization [16]. Building upon Proposition 9, it is interesting to observe that the stationary policy we are looking for in Section 5.2.1 is in some sense the steady-state distribution of the Lagrange multipliers. The intuition behind this solution is that the Lagrange multipliers act as interfaces between MN-DC and workload-power balance capturing all the resource availability and utility information which is relevant from a resource allocation point of view. However, to implement the aforementioned optimal resource allocation schemes, the optimal multipliers $\boldsymbol{\lambda}^*$ must be known. Toward this objective, we first review the celebrated stochastic approximation-based approaches or the Lyapunov optimization methods that are widely used in the stochastic network optimization and various related disciplines, and a novel dual learning approach is proposed in Section 5.3 to learn the optimal multipliers in both offline (training setting) and online (test set).

5.2.3 A review: Stochastic dual subgradient method

For the dual problem (5.13), a standard (sub-)gradient iteration involves taking the expectation over the distribution of $\mathbf{s}_t$ to compute the gradient [30]. This is challenging since the underlying distribution of $\mathbf{s}_t$ is usually unknown in practical applications. Even if the joint probability distribution functions were available, finding the expectations would be non-trivial.

To circumvent this challenge, a natural solution is resorting to stochastic approximation (SA) techniques [21, 54, 65]. The SDGD iterations can be written as

$$\boldsymbol{\lambda}_{t+1} \ = \Big[\boldsymbol{\lambda}_t + \mu \nabla \mathcal{D}_t(\boldsymbol{\lambda}_t)\Big]^+, \tag{5.16}$$

where the stochastic gradients $\nabla \mathcal{D}_t(\boldsymbol{\lambda}_t) = \mathbf{A}\mathbf{x}_t + \mathbf{c}_t$ are unbiased estimates of the true gradients, i.e., $\nabla \mathcal{D}(\boldsymbol{\lambda}_t) = \mathbb{E}[\nabla \mathcal{D}_t(\boldsymbol{\lambda}_t)]$. The primal variables $\mathbf{x}_t$ can be found by solving the following instantaneous sub-problems, one per t

$$\mathbf{x}_t \in \arg\min_{\mathbf{x}_t \in \mathcal{X}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t), \tag{5.17}$$

where the operator $\in$ accounts for cases that the Lagrangian has more than one minimizer. The minimization in (5.17) is not difficult to solve. For a number of relevant cost and utility functions, as revealed in Proposition 1, closed-form solutions for primal variables can be found. In this recursion,

the resource allocation decisions $\mathbf{x}_t$ are revealed "on the fly." To be more precise, they are functions of the dual iterates $\boldsymbol{\lambda}_t$ and state realizations $\mathbf{s}_t$. Moreover, given $\boldsymbol{\lambda}_t$, the next dual iterates $\boldsymbol{\lambda}_{t+1}$ depend only on the probability distributions of $\mathbf{s}_t$ via the stochastic gradients $\nabla\mathcal{D}_t(\boldsymbol{\lambda}_t)$. Consequently, the process $\{\boldsymbol{\lambda}_t\}$ is Markov with time-invariant transition probabilities, which entails the so-called virtual queue interpretation in the line of literatures [37, 38, 54, 72].

Due to the low complexity and robustness to non-stationary scenarios, SA-based approaches are widely used in various research disciplines; e.g., adaptive signal processing [40], stochastic network optimization [37, 38, 54], and energy management [72]. However, these methods usually suffer from the slow convergence speed, and do not explore many specific problem structures due to its universality. More importantly, the single-pass SA techniques, though efficient, waste a large number of valuable historical samples. Motivated by these facts, we aim to systematically design an offline-aided-online approach, which can significantly improve the online performance of SDGD in network resource allocation by utilizing the streaming big data, while preserving the merits of low complexity and fast adaptation.

5.3 Stochastic Network Optimization as A Learning Task

In Section 5.2, the dynamic resource allocation problem has been reformulated to a static stochastic programming, and the classical SDGD algorithm has been reviewed. However, bear in mind that our vision is to propose a novel offline-aided-online framework for this resource allocation problem, especially from a machine learning perspective, which is naturally tailored for the current era of big data. This expectation will be fulfilled in this section.

Before we put forward our proposed algorithms for this learning task, we proceed by first stating the assumptions we make throughout this chapter. These assumptions can typically be satisfied in online network resource allocation problems.

Assumption 1. *The network states $\mathbf{s}_t$ are independent and identically distributed (i.i.d.) over time t.*

Assumption 2. *The network costs $\Psi_t(\mathbf{x}_t)$ defined in (5.6) are σ-strongly convex, and their gradients are Lipschitz continuous with constant $\tilde{L}$, for all t.*

Assumption 3. *There exists a stationary policy $\boldsymbol{\chi}(\cdot)$ satisfying the Slater condition in the sense that $0 \leq \boldsymbol{\chi}(\boldsymbol{s}_t) \leq \bar{\mathbf{x}}$, for all $\mathbf{s}_t$, and $\mathbb{E}[\mathbf{A}\boldsymbol{\chi}(\boldsymbol{s}_t) + \mathbf{c}_t(\mathbf{s}_t)] \leq -\zeta$, where $\zeta > 0$ is the slack constant.*

Assumption 4. *The instantaneous dual function $\mathcal{D}_t(\boldsymbol{\lambda})$ defined in (5.21) is ϵ-strongly concave, and the associated gradient $\nabla\mathcal{D}_t(\boldsymbol{\lambda})$ is L-Lipschitz continuous, for all t.*

The condition in Assumption 1 requires the independence of random states over time, which is typical in the stochastic resource allocation in [26, 37, 38]. It is worth mentioning that this assumption is likely to be relaxed to ergodic setting. Assumption 2 ensures that the objective function is strongly convex, which is indeed a mild assumption, and many practical resource allocation problems with quadratic/exponential utility or cost functions will satisfy; e.g., [26]. Assumption 3 is the so-called slater condition in the stochastic setting, which is standard in the optimization literatures to ensure the existence of a bounded Lagrange multiplier [16]. Assumption 4 implies that the dual function is well-behaved. The L-Lipschitz continuous gradient $\nabla\mathcal{D}_t(\boldsymbol{\lambda})$ directly follows from the strongly-convex assumption in Assumption 2; i.e., $L = \rho(\mathbf{A}^\top\mathbf{A})/\sigma$, where $\rho(\mathbf{A}^\top\mathbf{A})$ is the spectral radius of the matrix $\mathbf{A}^\top\mathbf{A}$. The strong concavity is normally assumed in network optimization [47], and (if not equivalent) closely related to assumptions of the local smooth structure and the unique Lagrange multiplier in [26, 37, 38]. In the pessimistic case, it can be satisfied by subtracting an ℓ_2-regularizer in the dual function (5.12), and its sub-optimality can be analytically established in the next lemma.

Lemma 11. *Consider $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ as the optimal arguments of the primal and dual problems in (5.10) and (5.13), respectively. Further define the ℓ_2-regularized dual problem as*

$$\max_{\boldsymbol{\lambda}\geq 0} \mathcal{D}(\boldsymbol{\lambda}) - \frac{\epsilon}{2}\|\boldsymbol{\lambda}\|^2 \tag{5.18}$$

and consider $\hat{\boldsymbol{\lambda}}^*$ and $\hat{\mathbf{x}}^*$ as the optimal arguments of the dual problem (5.18) and its resultant primal problem. If the conditions in Assumptions 1 and 2 are satisfied, then for any $\epsilon > 0$, we have [cf. $\mathbf{x}^* = \{\mathbf{x}_t^*, \forall t\}$ and $\hat{\mathbf{x}}^* = \{\hat{\mathbf{x}}_t^*, \forall t\}$]

$$\mathbb{E}[\|\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*\|^2] \leq \frac{\epsilon}{2\sigma}\left(\|\boldsymbol{\lambda}^*\|^2 - \|\hat{\boldsymbol{\lambda}}^*\|^2\right),\ \forall\, \boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}^* \tag{5.19}$$

where $\boldsymbol{\Lambda}^*$ is the set of optimal dual variables for the original dual problem (5.13).

Proof. See the proof in Appendix A. □

The result in Lemma 11 characterizes an upper bound for the expected difference between the optimal arguments of the primal problem (5.10) and the primal problem for (5.18) in terms of the difference between the corresponding optimal dual arguments. In addition, by choosing a relatively small regularization parameter ϵ, the gap between the optimal arguments $\mathbf{x}_t^*$ and $\hat{\mathbf{x}}_t^*$ becomes smaller.

As an immediate consequence of Lemma 11, we have

$$\mathbb{E}[\|\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*\|] \leq \sqrt{\frac{\epsilon}{2\sigma}} \max_{\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}^*} \|\boldsymbol{\lambda}^*\| \tag{5.20}$$

which implies that the primal solution $\mathbf{x}^*$ will be $\mathcal{O}(\sqrt{\epsilon})$-optimal and feasible given the regularizer $\frac{\epsilon}{2}\|\boldsymbol{\lambda}\|^2$. And the sub-optimality in terms of the objective value can be easily captured using the Lipschitz gradient condition in Assumption 2. Clearly, the sub-optimality vanishes when ϵ approaches 0, or, the primal strong convexity constant σ is very large. Finally, note that as we eventually pursue a $\mathcal{O}(\mu)$-optimal online solution in Theorem 6, we will show that it suffices to set $\epsilon = \mathcal{O}(\mu)$ in our setting.

5.3.1 Batch learning via offline SAGA

In this subsection, we postulate a training setting for our network resource allocation problem by leveraging the historical data. To solve this *big data* problem in a feasible manner, an efficient batch learning approach is developed to obtain the empirical dual variables. We further show that the proposed learning approach with a simple iterate update has an order-optimal convergence rate.

Consider a training set with N historical state samples $\mathcal{S} := \{\boldsymbol{s}_n,\ 1 \leq n \leq N\}$ where the states $\boldsymbol{s}_n$ are defined in Section 5.2. The empirical version of the dual function (5.12) associated to the training set $\mathcal{S}$ with N samples can be written as

$$\hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}) := \frac{1}{N}\sum_{n=1}^{N} \hat{\mathcal{D}}_n(\boldsymbol{\lambda}) = \frac{1}{N}\sum_{n=1}^{N} \left[\min_{\mathbf{x}_n \in \mathcal{X}} \mathcal{L}_n(\mathbf{x}_n, \boldsymbol{\lambda}) \right]. \tag{5.21}$$

Note that the indices t of the instantaneous Lagrangian and dual functions in Section 5.2.2 have been changed to n to emphasize the dependence on the historical sample $\boldsymbol{s}_n$ and the fact that the empirical dual function is also defined on the historical sample set $\mathcal{S}$. Note that two dual functions

(5.12) and (5.21) are statistically equivalent when $N \to \infty$ according to the strong law of large number [39]. Consequently, the empirical dual problem can be expressed as

$$\max_{\boldsymbol{\lambda} \geq 0} \frac{1}{N} \sum_{n=1}^{N} \hat{\mathcal{D}}_n(\boldsymbol{\lambda}). \tag{5.22}$$

By recognizing that the objective function in (5.22) is decomposable as a sum of finite concave functions, it falls into a prevalent machine learning routine called empirical risk minimization (ERM) [75]. A natural way to solve (5.22) is to work directly with the full objective and use the gradient ascent iteration via

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k + \frac{\eta}{N} \sum_{n=1}^{N} \nabla \hat{\mathcal{D}}_n(\boldsymbol{\lambda}_k) \right]^+ \tag{5.23}$$

where the index k represents the batch learning index and η is the learning rate. While the batch gradient descent exhibits a decent convergence rate, its computational complexity will be prohibitively high when data size N is large. A typical alternative is to use again the stochastic gradient (SGD) method which randomly uses one gradient component per iteration. However, the SGD iteration for this setting only relies on the unbiased gradient measurements without fully exploring the finite sum structure of our batch learning problem. Specifically, the variance of SGD can only go to zero if decreasing step sizes are used, thus preventing a linear convergence rate like the batch gradient descent (5.23) and the SAGA that will be introduced later. A trade-off between stochastic methods, and batch gradient descent methods has been widely studied in machine learning literatures [24, 66].[1]

Leveraging this special problem structure, here we modify the recently developed Stochastic Average Gradient (SAGA) approach in [24] to our dual learning setting, and aim to efficiently compute an empirical Lagrange multiplier. Compared with the original SAGA setting in [24] for an unconstrained optimization problem, here we start from a constrained optimization problem (5.7), and are able to use SAGA by repeatedly reformulating the problem and using the Lagrange relaxation.

[1] It should be pointed out that the stochastic schemes for the empirical dual problem here are different from the one in Section 5.2.3, because the stochasticity here is introduced by the random selection of the algorithm itself, in contrast to the stochasticity of future system states in the online setting.

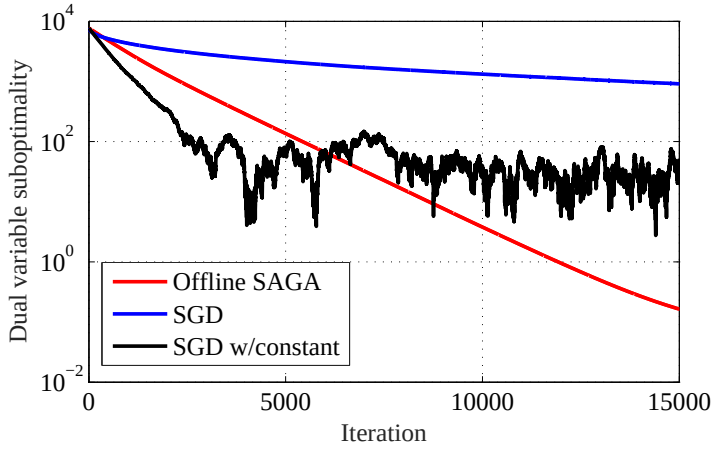

Figure 5.1: A comparison of convergence in the training setting (5.22) with 100 samples.

Per iteration k, the offline SAGA randomly evaluates one gradient component using the current iterate $\boldsymbol{\lambda}_k$. Therefore, the computational complexity of SAGA is comparable to SGD, and remarkably less than the gradient ascent (5.23). However, unlike SGD, SAGA maintains a table of the most recent gradient values $\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_k^n)$ for each component n, where $\boldsymbol{\pi}_k^n$ represents the previous iterate for which $\nabla\hat{\mathcal{D}}_n$ is evaluated. Though not up-to-date, this feature allows SAGA to incorporate a full gradient per iteration with respect to each function component $\hat{\mathcal{D}}_n$, thus achieving a faster convergence and low variance. The projected SAGA algorithm is summarized in Algorithm 6. Interestingly, like SGD, the offline SAGA also uses an unbiased gradient at each iteration k, namely

$$\mathbb{E}\left[\nabla\hat{\mathcal{D}}_\tau(\boldsymbol{\lambda}_k)-\nabla\hat{\mathcal{D}}_\tau(\boldsymbol{\pi}_k^\tau)+\sum_{n=1}^{N}\frac{\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_k^n)}{N}\right]=\sum_{n=1}^{N}\frac{\nabla\hat{\mathcal{D}}_n(\boldsymbol{\lambda}_k)}{N}$$

where the expectation is over the random selection τ.

Under Assumption 4, the convergence rate of the offline SAGA is established in the next theorem.

Theorem 3. *Consider $\boldsymbol{\lambda}_S^*$ as the optimal argument of the dual problem in (5.22). If the stepsize is chosen as $\eta = 1/(3L)$ with the Lipschitz constant L defined in Assumption 4, then SAGA in*

Algorithm 6 Offline SAGA Iteration for batch learning

1: **Initialize:** with a proper $\boldsymbol{\lambda}_0$, $\boldsymbol{\pi}_0^n = \boldsymbol{\lambda}_0$, $\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_0^n),\ \forall n$, and stepsize η.

2: **for** $k = 0, 1, 2 \ldots$ **do**

3: Pick an τ uniformly at random from set $\{1, \ldots, N\}$.

4: Compute $\nabla\hat{\mathcal{D}}_\tau(\boldsymbol{\lambda}_k)$ and update dual variable $\boldsymbol{\lambda}_{k+1}$ via

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k + \eta\left(\nabla\hat{\mathcal{D}}_\tau(\boldsymbol{\lambda}_k) - \nabla\hat{\mathcal{D}}_\tau(\boldsymbol{\pi}_k^\tau) + \sum_{n=1}^{N}\frac{\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_k^n)}{N}\right)\right]^+. \tag{5.24}$$

5: Store the gradient components $\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_{k+1}^n),\ \forall n$, with

$$\boldsymbol{\pi}_{k+1}^n := \begin{cases} \boldsymbol{\lambda}_k, & \text{if } n = \tau \\ \boldsymbol{\pi}_k^n, & \text{otherwise.} \end{cases} \tag{5.25}$$

6: **end for**

Algorithm 1 achieves the linear convergence rate, namely

$$\mathbb{E}\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{\mathcal{S}}^*\|^2 \leq \left(1 - \frac{1}{\alpha}\right)^k \left[\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_{\mathcal{S}}^*\|^2 - \frac{2N}{3L}\left[\hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_0) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\mathcal{S}}^*) - \langle\nabla\hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\mathcal{S}}^*), \boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_{\mathcal{S}}^*\rangle\right]\right] \tag{5.26}$$

where $\frac{1}{\alpha} = \min(\frac{1}{4N}, \frac{1}{3\kappa})$*, and the expectation is taken over all choices of index* τ *up to iteration* k*.*

Proof. The proof can follow the lines of [24] with the proximity operator replaced by the projection operator. Due to limited space, we omit the proof in this version, but will include it upon editor's request. □

Theorem 3 entails that the sequence of iterates $\boldsymbol{\lambda}_k$ generated by SAGA converge exponentially to the empirical Lagrange multiplier $\boldsymbol{\lambda}_{\mathcal{S}}^*$ in expectation. Using the L-smooth property of the dual function in Assumption 4, we can further derive the sub-optimality bound in terms of function values as

$$\mathbb{E}[\hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\mathcal{S}}^*) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_k)] \leq L\mathbb{E}\|\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{\mathcal{S}}^*\|^2 \leq (1 - 1/\alpha)^k LC_{\mathcal{S}}$$

where $C_{\mathcal{S}}$ is the initial error in the RHS of (5.26). Remarkably, the offline SAGA for our training problem is able to obtain the order-optimal convergence rate among all first-order approaches at the cost of only one gradient evaluation per iteration. For a better illustration, we compare the convergence of the offline SAGA with two SGD variants in Fig. 5.1: SGD represents the stochastic gradient ascent with diminishing stepsizes $\eta = 1/\sqrt{k}$; and SGD w/constant denotes the stochastic gradient ascent with constant stepsizes $\eta = 0.2$. It can be observed that SAGA is the only stochastic method that converges linearly to the optimal argument.

Remark 5. As the offline SAGA exhibits a competitive performance in terms of convergence rate and computational complexity, one may attempt to implement the empirical Lagrange multiplier from the training setting directly to the test scenarios, as a typical machine learning procedure. However, this is not applicable in the network optimization setting, because it will turn off adaptation and learning and lose the ability to track system dynamic variations, thus the queue stability can not be guaranteed once the system statistics change. Motivated by this limitation, we are targeting an offline-aided-online scheme, which incorporates the benefit of offline training to mitigate the online stochasticity, but still preserves the online adaptation capability.

5.3.2 Learning-while-testing via online SAGA

While the offline SAGA in Section. 5.3.1 is well-suited for the offline learning from the batch dataset to initiate a hot start for SDGD iteration (5.16), a novel dynamic resource allocation approach (called online SAGA) is proposed in this section which operates in a *LeArning-while-TestINg* (Latin) fashion. The online SAGA can be viewed as a combination of *stochastic approximation* and *statistical learning*: in the learning phase, it preserves the simple update of offline SAGA and maintains the learning error always below the statistical accuracy; in the testing phase, it incorporates the merits of SDGD to track queue variations and guarantee the queue stability. The online SAGA is summarized in Algorithm 7.

The online SAGA in Algorithm 7 consists of two complementary stages: offline training and online testing. In the offline training, Algorithm 7 runs $\hat{T} = KN^{\text{off}}$ SAGA iterations in (5.24) on a training set with N^{off} historical samples. With the output of offline training as a hot start, the online stage incorporates the Latin procedure to keep learning from fresh data and track queue variations.

Specifically, instead of directly using the empirical dual variable for resource allocation, SAGA solves $\min_{\mathbf{x}_t \in \mathcal{X}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\gamma}_t)$ by constructing an effective dual variable $\boldsymbol{\gamma}_t$ as

$$\boldsymbol{\gamma}_t = \boldsymbol{\lambda}_t + \mu \mathbf{Q}_t - \boldsymbol{\theta}, \ \forall t. \tag{5.27}$$

Notice that $\boldsymbol{\gamma}_t$ is a linear combination of the empirical dual variable $\boldsymbol{\lambda}_t$ and the instantaneous queue length $\mathbf{Q}_t$, where the control variable μ tunes the weights of these two factors, and the constant $\boldsymbol{\theta}$ controls the steady-state behavior of $\boldsymbol{\gamma}_t$, as will be specified in Theorems 5-6 and the discussion therein. Clearly, unlike SDGD, the online SAGA is able to leverage an online learning phase, where the algorithm runs K iterations of the offline SAGA in a dynamic training set including the new sample $\mathbf{s}_t$.

For a better understanding of the design philosophy, we first compare the offline and online SAGA so as to motivate the importance of the Latin strategy used in our proposed online SAGA. As mentioned earlier, the offline SAGA is designed for the batch learning from a given training dataset, which is well-motivated with its low computational complexity and guaranteed linear convergence rate to the exact optimal point. However, one cannot naturally equate a good learning algorithms with a fast and accurate optimization algorithm for ERM tasks. And for many practical applications, an approximate solution of ERM is sufficient, since the ultimate goal is to solve expected risk minimization (e.g., (5.13)), or, from a learning perspective, to avoid overfitting and preserve a desired generalization capability in the test dataset (in the online resource allocation). Especially for a massive dataset, obtaining a precise solution in ERM is not that wise.

To this end, a natural question will be how accurate is good enough in ERM, while considering the trade-off between the statistical and optimization accuracy. The main idea of the online SAGA in the learning phase is to incrementally increase the size of the training set in a way that the optimization error of the optimization variable is just below the statistical accuracy for the current dataset [23]. Consequently, the online SAGA algorithm only needs several gradient evaluations per new datum to maintain a sufficiently low optimization error.

For the remaining derivations of the online SAGA, it is instructive to formally state the definition of the uniform convergence bound that upper bounds the difference between the empirical risk in

Algorithm 7 Online SAGA for Learning-while-Testing

1: **Offline initialize:** size of N^{off} historical samples $\mathcal{S} = \{\mathbf{s}_n\}$, the dual variable $\hat{\boldsymbol{\lambda}}_0$, the intermediate variables $\hat{\boldsymbol{\pi}}_0^n$, $\nabla\hat{\mathcal{D}}_n(\hat{\boldsymbol{\pi}}_0^n),\ \forall n$, and stepsize η.

2: **Offline phase:** for the historical samples $\{\mathbf{s}_n\}_{N^{\text{off}}}$, run offline SAGA for $\hat{T} = KN^{\text{off}}$ iterations.

3: **Online initialize:** initiate a hot start $\boldsymbol{\lambda}_0$, $\boldsymbol{\pi}_0^n$, $\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_0^n),\ 1 \leq n \leq N^{\text{off}}$ from the output of the offline SAGA, a dynamic dataset $\mathcal{S}$ including N^{off} historical samples, the queue length $\mathbf{Q}_0$, and the control variables $\mu > 0$ and $\boldsymbol{\theta} = \sqrt{\mu}\log^2(\mu)$.

4: **for** $t = 0, 1, 2\ldots$ **do**

5: **Online testing phase:**

6: Construct the effective dual variable $\boldsymbol{\gamma}_t = \boldsymbol{\lambda}_t + \mu\mathbf{Q}_t - \boldsymbol{\theta}$, observe the system state $\mathbf{s}_t$, and obtain the decision $\mathbf{x}_t$ by the closed-form solutions (5.14)-(5.15), or equivalently, solving the instantaneous Lagrangian minimization

$$\min_{\mathbf{x}_t \in \mathcal{X}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\gamma}_t). \tag{5.28}$$

7: Update the queues $\mathbf{Q}_{t+1}$ via the dynamics (5.1)-(5.2).

8: **Online learning phase:**

9: Add the new sample $\mathbf{s}_t$ to the set $\mathcal{S}$ with $N_{t+1} = N_t + 1$, and initialize a gradient component $\nabla\hat{\mathcal{D}}_{N_{t+1}}(\hat{\boldsymbol{\lambda}}_0)$.

10: Run K SAGA iterations (5.24) in the training set $\mathcal{S}$.

11: Update the empirical dual variable $\boldsymbol{\lambda}_{t+1}$ and the gradients $\nabla\hat{\mathcal{D}}_n(\boldsymbol{\pi}_{t+1}^n),\ \forall n$, via the output of step 10.

12: **end for**

(5.21) and the statistical loss in (5.12) for all $\boldsymbol{\lambda}$ with high probability (w.h.p)

$$\sup_{\boldsymbol{\lambda} \geq 0} |\mathcal{D}(\boldsymbol{\lambda}) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda})| \leq \mathcal{H}_s(N),\ w.h.p \tag{5.29}$$

where $\mathcal{H}_s(N)$ is a constant, depending on N which is the number of samples in the training set $\mathcal{S}$ [75]. In many practical cases, $\mathcal{H}_s(N)$ is proportional to $\sqrt{d/N}$ under some regularity conditions, or proportional to d/N in a more strict function classes [75, Section 3.4], where d is the capacity of the function class (e.g., VC dimension) where the weight vector $\boldsymbol{\lambda}$ belongs to. Now let $\boldsymbol{\lambda}_{\mathcal{S}}$ denote

an $\mathcal{H}_o(N)$-optimal solution obtained by the offline SAGA in the training set $\mathcal{S}$ with sample size N, where $\mathcal{H}_o(N)$ is the optimization accuracy; i.e., $\mathbb{E}[\hat{\mathcal{D}}_{\mathcal{S}}^* - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\mathcal{S}})] \leq \mathcal{H}_o(N)$ with the optimal argument $\boldsymbol{\lambda}_{\mathcal{S}}^*$. Clearly, the difference the training and the testing settings will be approximately bounded by

$$\mathcal{D}(\boldsymbol{\lambda}^*) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\mathcal{S}}) \leq \mathcal{H}_o(N) + \mathcal{H}_s(N), \ w.h.p \tag{5.30}$$

which is the summation of the statistical error $\mathcal{H}_s(N)$ and the optimization error $\mathcal{H}_o(N)$. Therefore, it is not necessary and computational efficient to find a better approximation of the minimizer in the training set once $\mathcal{H}_o(N) \leq \mathcal{H}_s(N)$. In other words, an algorithm solves the ERM problem to within its statistical accuracy is good enough, which motivates the design of the online SAGA approach.

Observe that in the offline SAGA, the optimization error decreases exponentially with an *approximate* rate $\Gamma_N = 1 - \min(1/N, 1/\kappa)$ [cf. Theorem 3]. Hence, for a small N, the constant $1 - 1/\kappa$ dominates the convergence rate; for a big N, the rate depends on $1 - 1/N$. Targeting a efficient solution for learning from massive dataset, we assume the so-called *big-data condition* satisfied; i.e., the sample size is large enough so that $N > \kappa$. Implementing the idea of dynamic SAGA [23] in our learning phase, we are able to achieve a desired optimization error $\mathcal{H}_o(N) \leq \mathcal{H}_s(N)$ after only $2N$ iterations in total, given the current sample size N.

5.3.3 Optimality and stability analysis

Coupled with online resource allocation tasks, analyzing the online SAGA is challenging. In this section, we will first analyze the performance of the statistical learning, from which we derive the results for online testing, and queue stability.

For the online learning phase in Algorithm 7, we can establish the upper bound for the optimization error.

Lemma 12. *Consider the online SAGA in Algorithm 7, and the dynamic training set $\mathcal{S}$ at time t with size $N_t := t + N^{\text{off}}$. For the statistical accuracy $\mathcal{H}_s(N) = \tilde{d}/N^{-\beta}$, with a constant $\tilde{d} > 0$ and $0 < \beta \leq 1$, if we select $K \geq 2$ and $N^{\text{off}} = \kappa$, then the empirical learning error of the online SAGA at time slot t is upper bounded by*

$$\mathbb{E}\left[\hat{\mathcal{D}}_{\mathcal{S}}^* - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_t)\right] \leq c\mathcal{H}_s(N_t) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_t}\right)^K \tag{5.31}$$

where the constant ξ is defined as $\xi = 4\kappa(\mathcal{D}^* - \mathcal{D}(\hat{\boldsymbol{\lambda}}_0))$ with the initial $\hat{\boldsymbol{\lambda}}_0$, and the constant c satisfies

$$c = \left(\sum_{k=1}^{K-1} \left(\frac{N_t + 1}{N_t} \right)^k \right)^{-1} \leq 1, \ \forall t. \tag{5.32}$$

Proof. See the proof in Appendix B. □

Lemma 12 fits our intuition. Specifically, for a given sample size N_t, increasing the number of per-slot SAGA iterations K will lead to a smaller empirical learning error. As the computational complexity of online SAGA algorithm will increase as K increases, we are ready to show next that running SAGA two iterations per new sample suffices to guarantee the optimization error below the statistical accuracy.

Corollary 1. *Under the same assumptions as Lemma 12, if $K = 2$ and $N^{\text{off}} = \kappa$, then the learning error satisfies that*

$$\mathbb{E}\left[\hat{\mathcal{D}}_{\mathcal{S}}^* - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_t) \right] \leq \frac{N_t}{N_t + 1} \mathcal{H}_s(N_t) + \frac{\xi}{e^2} \left(\frac{\kappa}{N_t} \right)^2. \tag{5.33}$$

Corollary 1 shows when the sample size N_t is sufficiently large, that the optimization accuracy with only $K = 2$ learning iterations per slot (per new sample) will approach the statistical accuracy.

Notice that the online SAGA learning phase maintains a dynamic training set with size N_t at online time slot t, and iterates only K offline SAGA updates for each new datum. This is largely orthogonal to the learning protocol in [37], where a histogram needs to be built to approximate the empirical distribution of the underlying random states and a large-scale deterministic optimization problem has to be solved per time slot. In the online SAGA framework, we theoretically only need N^{off} batch samples and run K iterations per slot to ensure the optimization error below the statistical accuracy. In practice, we can further skip the batch learning step in the sense that we do not need any a-priori system statistical knowledge, and the simulations validates that directly operating the online Latin procedure will still significantly improve the online performance of stochastic resource allocation approaches.

We next proceed by stating the asmptotical convergence of the empirical dual variables.

Theorem 4. *Consider the online SAGA in Algorithm 7. If we choose the historical sample size as $N^{\text{off}} = \kappa$, and the learning iteration $K \geq 2$, then the empirical dual variables satisfy that*

$$\lim_{t\to\infty} \boldsymbol{\lambda}_t = \boldsymbol{\lambda}^*, \quad w.p.1 \tag{5.34}$$

where the $\boldsymbol{\lambda}^*$ is the optimal dual variable for the expected dual function minimization (5.13).

Proof. Using the uniform convergence bound (5.29) and Lemma 12, it follows that

$$|\mathcal{D}(\boldsymbol{\lambda}^*) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_t)| \leq (1+c)\mathcal{H}_s(N_t) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_t}\right)^K. \tag{5.35}$$

And the lemma then follows by using the fact that the dual function is strongly-concave, and $\lim_{t\to\infty} \mathcal{D}(\boldsymbol{\lambda}^*) - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_t) \leq \lim_{t\to\infty}(1+c)\mathcal{H}_s(N_t) + \xi/e^K(\kappa/N_t)^K = 0,\ w.p.1.$ □

Theorem 4 ensures that the empirical dual variable learned by the online SAGA converges to the optimal dual variable $\boldsymbol{\lambda}^*$ with probability 1 (w.p.1) even for small N^{off} and K [23].

Remark 6. The exact convergence to the Lagrange multiplier in Theorem 4 essentially requires infinite sample size of training set as $t \to \infty$. While this statement is mathematically concise and elegant, various ways, from a practical perspective, are promising to tackle the learning step within a budgeted training set. Consider a case where the training set has a fixed sample size budget $W > \kappa$, and the online SAGA always keeps the most recent W samples with their associated gradients in the training set. Following Lemma 12 and [52, Lemma 6], the error term in (5.35) will still stay at the level $\delta' = \mathcal{O}(\mathcal{H}_s(W))$ for a sufficiently large W, thus the empirical dual iterates will converge to a δ'-neighborhood of the optimal Lagrange multiplier. However, one can easily follow the steps of Theorems 5 and 6 to show that this δ'-inexactness will not affect the arguments (5.37)-(5.39) whenever W is sufficiently large such that $\delta' = \mathbf{o}(\min\{\mu, \boldsymbol{\theta}\})$.

Building upon the convergence of empirical dual variable in the statistical learning, the ideal next step is to show that the effective dual variable $\boldsymbol{\gamma}_t$ also converges the optimal dual variable of (5.13) thus the online resource allocation $\mathbf{x}_t$ is also asymptotically optimal, as is the function of the effective dual variable (9). However, directly showing the convergence of $\boldsymbol{\gamma}_t$ is nontrivial. To see this point, rewrite the recursion of $\boldsymbol{\gamma}_t$ as

$$\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t + (\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}_t) + \mu(\mathbf{Q}_{t+1} - \mathbf{Q}_t),\ \forall t \tag{5.36}$$

where the dynamics of $\boldsymbol{\gamma}_t$ depends on the variations of $\boldsymbol{\lambda}_t$ and $\mathbf{Q}_t$. Accordingly, we will first study the asymptotical behavior of queue lengths, and then show that $\boldsymbol{\gamma}_t$ converges to the neighborhood of $\boldsymbol{\lambda}^*$ via (5.34) and (5.36).

Define a time-varying target variable $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\lambda}^* - \boldsymbol{\lambda}_t + \boldsymbol{\theta}$, which is the dual variable optimality residual $\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_t$ plus the control variable $\boldsymbol{\theta}$. Based on Theorem 4, it readily follows that $\lim_{t\to\infty} \tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta},\ w.p.1$. Next we show that the queue length $\mathbf{Q}_t$ will be attracted towards the time-varying target $\tilde{\boldsymbol{\theta}}_t/\mu$, and further show the stability of the steady-state queue lengths.

Lemma 13. *Consider $\mathbf{Q}_t$ and μ as the queue lengths and the control variable in the online SAGA, respectively. There exist a constant $B = \Theta(\frac{1}{\sqrt{\mu}})$ depending on μ, and a finite time $T_B < \infty$, such that for all $t \geq T_B$, if $\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\| > B$, it holds w.p.1 that*

$$\mathbb{E}\left[\left\|\mathbf{Q}_{t+1} - \tilde{\boldsymbol{\theta}}_t/\mu\right\| \middle| \mathbf{Q}_t\right] \leq \left\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\right\| - \sqrt{\mu}. \tag{5.37}$$

Proof. See the proof in Appendix C. □

The result of Lemma 13 characterizes the so-called drift behavior of the queue length $\mathbf{Q}_t$. Intuitively, when the queue length deviates from the target $\tilde{\boldsymbol{\theta}}_t/\mu$, it will be bounced back to the target at next slot. Upon establishing the drift behavior of the queues, we are on the track to show the asymptotical queue lengths.

Theorem 5. *Consider again $\mathbf{Q}_t, \boldsymbol{\theta}, \mu$ as the queue lengths and the control variables in the online SAGA. Under the online SAGA, the steady-state queue length satisfies*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\mathbf{Q}_t\right] = \frac{\boldsymbol{\theta}}{\mu} + \mathcal{O}(\frac{1}{\sqrt{\mu}}). \tag{5.38}$$

Proof. See the proof in Appendix D. □

Theorem 5 implies that the steady-state queue lengths will stay close to $\boldsymbol{\theta}/\mu$, within a distance of $\mathcal{O}(1/\sqrt{\mu})$. As a result, a smaller $\boldsymbol{\theta}$ will lead to a lower average delay, and one may attempt to set the control variable close to zero to ensure a sufficiently low delay. Though attractive, a small $\boldsymbol{\theta}$ will guide the effective dual variable to a value slightly larger than $\boldsymbol{\lambda}^*$ in the average sense, since the queue lengths $\mathbf{Q}_t$ are always nonnegative. Hence, a small $\boldsymbol{\theta}$ will lead to an additional optimality

loss. Using the similar arguments in [37, 38], we next show that by appropriately setting $\boldsymbol{\theta}$, the online SAGA is still asymptotically optimal.

Theorem 6. *Consider Ψ^* the optimal objective value of (5.7) under any feasible control with all future information. If the control variable is chosen as $\boldsymbol{\theta} = \sqrt{\mu}\log^2(\mu)$, then with an appropriate μ, the proposed online SAGA algorithm yields a near-optimal solution for (5.7) in the sense that*

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\Psi_t\left(\mathbf{x}_t(\boldsymbol{\gamma}_t)\right)\right] \leq \Psi^* + \mathcal{O}(\mu) \tag{5.39}$$

where $\mathbf{x}_t(\boldsymbol{\gamma}_t)$ denotes the real-time operations obtained from the Lagrangian minimization (5.28).

Proof. See the proof in Appendix E. □

Combining Theorems 5 and 6, it turns out that by setting the control variable as $\boldsymbol{\theta} = \sqrt{\mu}\log^2(\mu)$, the online SAGA is asymptotically optimal with an average queue length $\mathcal{O}(\frac{1}{\sqrt{\mu}}\log^2(\mu))$. The results imply that the proposed SAGA approach is able to achieve a near-optimal $[\mu, \frac{1}{\sqrt{\mu}}\log^2(\mu)]$ cost-delay tradeoff in the context of stochastic network optimization [54]. Comparing with the standard $[\mu, \frac{1}{\mu}]$ tradeoff of the SDGD or the well-known backpressure algorithm, it reveals that the offline-aided-online design can significantly improve the online performance in terms of the delay. Our result is superior under the current setting in the sense that the "optimal" tradeoff $[\mu, \log^2(\mu)]$ in [37] is derived under a more "strict" local polyhedral assumption, while it is not asserted in our context.

Remark 7. Readers familiar with literatures on statistical learning and stochastic network optimization can recognize similarities and differences between the online SAGA and the various algorithms proposed in therein. Fundamentally, the Latin design of the online SAGA amounts to choosing an appropriate online policy, or equivalently an effective dual variable, which satisfies the following criterion.

P1) The effective dual variable should be initiated or adjusted close enough to the optimal Lagrange multiplier so that the online algorithm will almost operate in an optimal resource allocation strategy [cf. Proposition 9]. While the traditional SDGD uses incremental update to adjust dual variables, the online SAGA could obtain a near-optimal effective dual variable much faster than SDGD, thanks to the contribution of statistical learning. This fact reduces the transient time of the

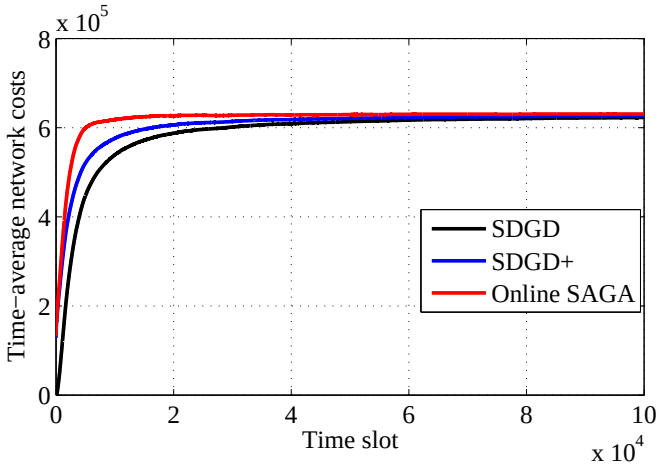

Figure 5.2: Comparison of time-average network costs. ($I = J = 4$, $N^{\text{off}} = 1000$, $K = 2$)

Markov chain $\mathbf{Q}_t$, and the convergence time of the online SAGA, thus the reduced delay can be expected.

P2) On the other hand, the online SAGA should not lose the control of queue lengths. Hence, the effective dual variable should also sensitively "touch" queue evolutions to guarantee the long-term queue stability, which necessitates the leverage of stochastic approximation for instantaneous measurements of queue update.

Following these principles, the proposed Latin framework is likely to be extended to settings where we use second-order learning methods in learning phase (e.g., addNewton in [52]), and momentum-based approaches in testing phase [47].

5.4 Numerical Tests

This section presents numerical tests to confirm the analytical claims, and demonstrate the merits of the proposed approach. The network considered in this section has $I = 4$ DCs and $J = 4$ MNs. Performance is tested in terms of the time-average of instantaneous network cost

$$\Psi_t(\mathbf{x}_t) := \sum_{i\in\mathcal{I}} \alpha_{i,t} \left(e_{i,t} x_{i,t}^2 - P_{i,t}^{\mathrm{r}}\right) + \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c_{i,j}^{\mathrm{d}} \tilde{x}_{i,j,t}^2 \tag{5.40}$$

where the energy transaction price $\alpha_{i,t}$ is uniformly distributed within $[10, 30]$ \$/kWh; the energy efficiency factors are considered time-invariant with values $\{e_{i,t}\} = \{1.2, 1.3, 1.4, 1.5\}$; samples

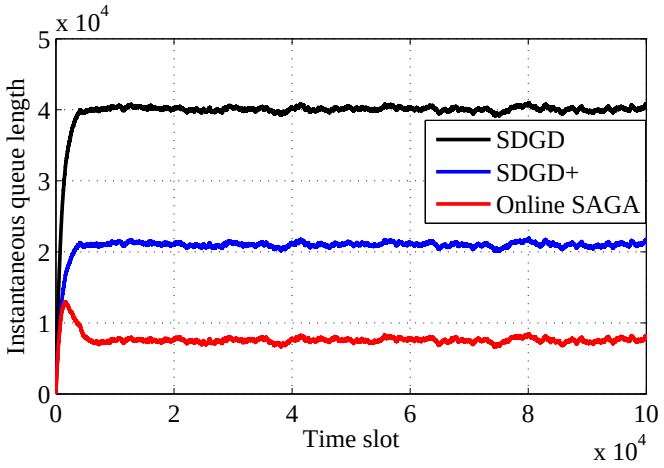

Figure 5.3: Comparison of network queue lengths. ($I = J = 4$, $N^{\text{off}} = 1000$, $K = 2$)

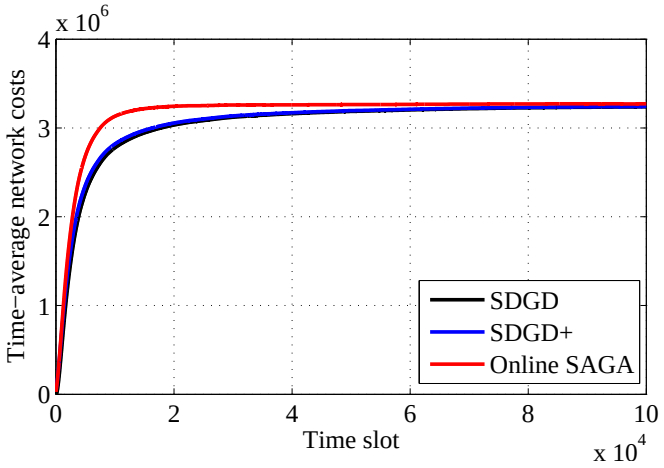

Figure 5.4: Comparison of time-average network costs. ($I = J = 20$, $N^{\text{off}} = 1000$, $K = 2$)

of the renewable supply $\{P^{\text{r}}_{i,t}\}$ are generated from a uniform distribution within $[10, 50]$ kWh; and the bandwidth cost is set to $c^{\text{d}}_{i,j} = 40/B_{i,j}$, with bandwidth limits $\{B_{i,j}\}$ generated from a uniform distribution with support $[10, 100]$. According to workloads, the computing capacities at each DC are set $\{M_i\} = \{200, 150, 100, 100\}$, delay-tolerant workloads $\{v_{j,t}\}$ arrive at each MN j according to a uniform distribution within $[10, 150]$ workload unit. Finally, the control variables are chosen as $\boldsymbol{\theta} = \sqrt{\mu}\log^2(\mu)$ and $\mu = 0.1$ by default, otherwise will be stated. We introduce two alternatives to benchmark the proposed online SAGA approach: **SDGD** is the classical stochastic dual subgradient algorithm (see e.g., [21, 54]), and **SDGD+** is an advanced version of SDGD in the sense that it uses a hot start obtained by the offline SAGA with N^{off} samples as well, but maintains the same online

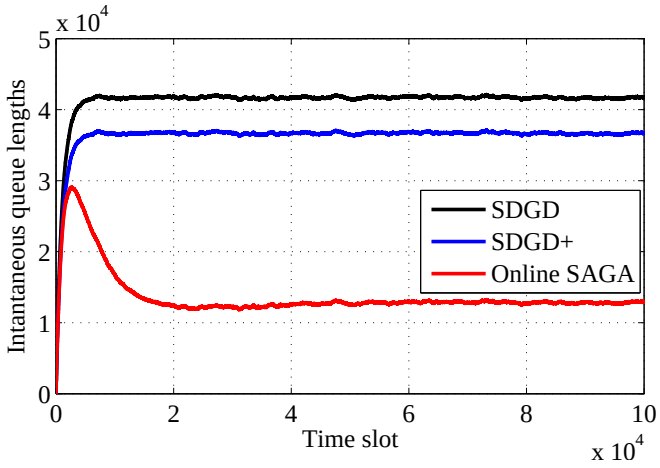

Figure 5.5: Comparison of network queue lengths. ($I = J = 20$, $N^{\text{off}} = 1000$, $K = 2$)

operations.

5.4.1 A comparison of online performance

Performance is first compared with moderate training sample size $N^{\text{off}} = 1000$ in Figs. 5.2-5.3. For the network cost, the three algorithms converge to the same value, and the online SAGA has the fastest convergence speed since it will quickly achieve the optimal operating phase by leveraging the learning power. In addition, leveraging the Latin procedure, the online SAGA incurs a much lower delay as the queue size is only 40% of that under SDGD+ and 20% of that under SDGD. Clearly, the offline training can improve the delay performance of SDGD+, and the online SAGA with Latin procedure will further gain relative to SDGD+. These two metrics are further compared in Figs. 5.4-5.5 over a large network with $I = 20$ DCs and $J = 20$ MNs. While three algorithm exhibit the similar performance in terms of network cost, the delay of the online SAGA is still much lower than the alternatives. Notice that the delay performance of SDGD+ closes to that of SDGD compared with the gap in Fig. 5.3, which indicates that the training samples $N^{\text{off}} = 1000$ may be not enough to learn a good initial point for SDGD+ over such a large network.

5.4.2 An improved trade-off of cost and network delay

For a fair comparison to SDGD, the tradeoffs of the control variable μ, $\boldsymbol{\theta}$, and per-slot learning iteration K are studied in Figs. 5.7-5.9 with no a-priori training samples; i.e., $N^{\text{off}} = 0$. For all

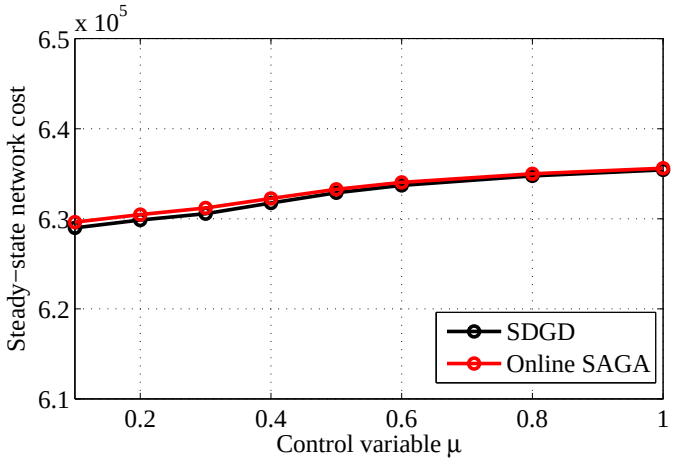

Figure 5.6: A tradeoff of the control variable μ in terms of network cost. ($I = J = 4$, $N^{\text{off}} = 0$, $K = 2$)

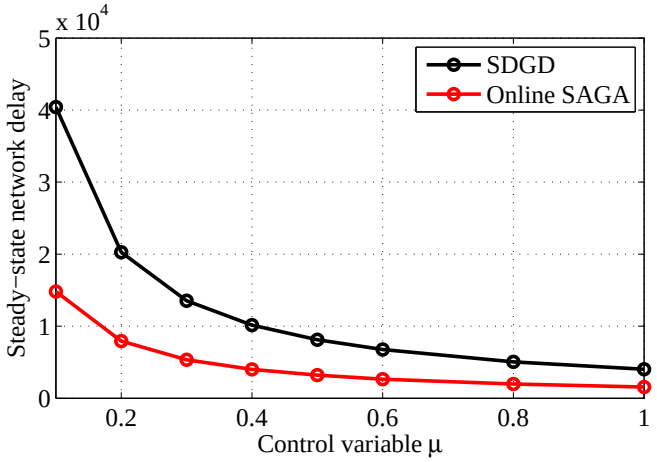

Figure 5.7: A tradeoff of the control variable μ in terms of network queue lengths. ($I = J = 4$, $N^{\text{off}} = 0$, $K = 2$)

the choices of μ, the online SAGA still obtains a much smaller delay with a similar network cost relative to SDGD [cf. Fig. 5.7], and the delay increases much slower when μ decreases [cf. Fig. 5.6], thanks to a better delay-cost tradeoff $[\mu, \frac{1}{\sqrt{\mu}}\log^2(\mu)]$. As shown by Lemma 12, the per-slot learning error will decrease as the learning iteration K increases. Finally, Figs. 5.8-5.9 represent the tradeoff of the performance and the learning complexity in terms of K. It is observable that increasing K will slightly improve the convergence speed, and significantly reduce the delay at the expense of a higher computational complexity.

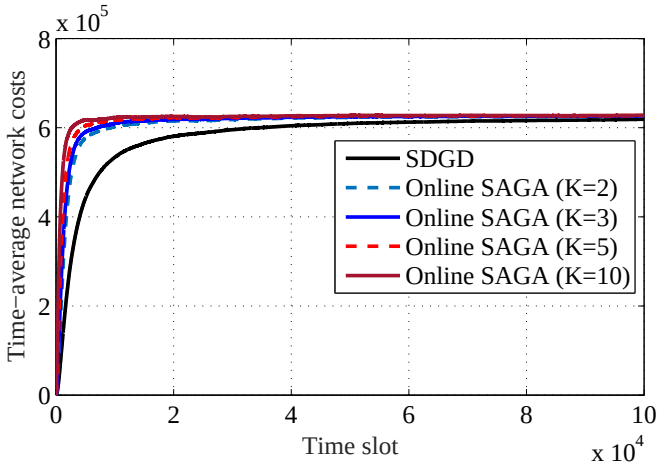

Figure 5.8: A comparison of different K in terms of network cost. ($I = J = 4$, $N^{\text{off}} = 0$)

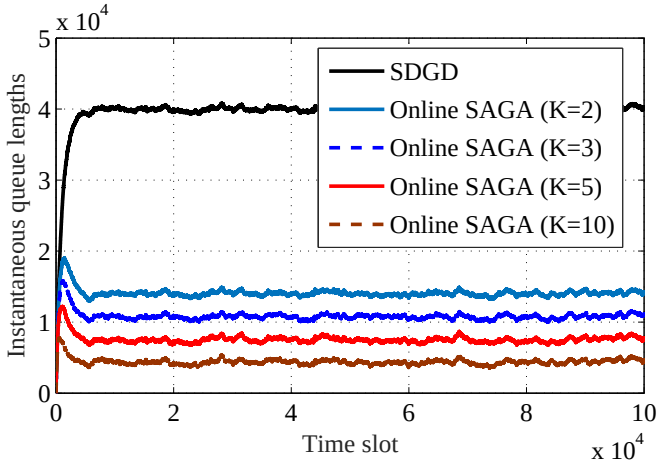

Figure 5.9: A comparison of different K in terms of network delay. ($I = J = 4$, $N^{\text{off}} = 0$)

5.5 Appendices

5.5.1 Proof of Lemma 1

The proof follows the steps in [41, Lemma 3.2]. Define the approximate Lagrangian function as $\hat{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda}) := \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) - \frac{\epsilon}{2}\|\boldsymbol{\lambda}\|^2$. From the definition, it readily follows that $(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\lambda}}^*)$ is a saddle point for the approximate Lagrangian function $\hat{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda})$, while $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is a saddle point for the original Lagrangian function $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$; i.e.,

$$\hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \boldsymbol{\lambda}) \leq \hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\lambda}}^*) \leq \hat{\mathcal{L}}(\mathbf{x}, \hat{\boldsymbol{\lambda}}^*), \ \forall \mathbf{x} \in \mathcal{X}, \ \boldsymbol{\lambda} \geq 0.$$

Plugging $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, we obtain

$$\begin{aligned} 0 &\leq \hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\lambda}}^*) - \hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \boldsymbol{\lambda}^*) \\ &= (\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] - \frac{\epsilon}{2}\|\hat{\boldsymbol{\lambda}}^*\|^2 + \frac{\epsilon}{2}\|\boldsymbol{\lambda}^*\|^2. \end{aligned}$$

We next consecutively upper bound the terms $(\hat{\boldsymbol{\lambda}}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t]$ and $-(\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t]$. First, we have $\mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] = \mathbb{E}[\mathbf{A}\mathbf{x}_t^* + \mathbf{c}_t] + \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \leq \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)]$, where the inequality follows that $\mathbf{x}^*$ is a feasible solution to the original problem so that $\mathbb{E}[\mathbf{A}\mathbf{x}_t^* + \mathbf{c}_t] \leq 0$. Multiplying each sides by the entry-wise nonnegative $\hat{\boldsymbol{\lambda}}^*$, and using the definition of $\hat{\mathcal{L}}(\mathbf{x}, \boldsymbol{\lambda})$, we arrive at

$$\begin{aligned} (\hat{\boldsymbol{\lambda}}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] &\leq (\hat{\boldsymbol{\lambda}}^*)^\top \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \\ &= \nabla_{\mathbf{x}} \hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\lambda}}^*)^\top \mathbb{E}[\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*] - \mathbb{E}[\nabla \Psi_t(\hat{\mathbf{x}}_t^*)^\top (\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \\ &\leq -\mathbb{E}[\nabla \Psi_t(\hat{\mathbf{x}}_t^*)^\top (\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \end{aligned}$$

where the inequality comes from the optimality condition for minimization over the approximate Lagrangian function; i.e., $\nabla_{\mathbf{x}} \hat{\mathcal{L}}(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\lambda}}^*)^\top \mathbb{E}[\mathbf{x}_t - \hat{\mathbf{x}}_t^*] \geq 0, \ \forall \mathbf{x}_t \in \mathcal{X}$. Likewise, we have $-\mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] = -\mathbb{E}[\mathbf{A}\mathbf{x}_t^* + \mathbf{c}_t] - \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)]$. Multiplying each sides by $\boldsymbol{\lambda}^*$, it follows that

$$\begin{aligned} -(\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] &= -(\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\mathbf{x}_t^* + \mathbf{c}_t] - (\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \\ &\overset{(a)}{=} -(\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}(\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*)] \\ &= \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*] - \mathbb{E}[\nabla \Psi_t(\mathbf{x}_t^*)^\top (\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*)] \\ &\overset{(b)}{\leq} -\mathbb{E}[\nabla \Psi_t(\mathbf{x}_t^*)^\top (\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*)] \end{aligned}$$

where the equality (a) uses the complementary slackness for the saddle point $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ so that $(\boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\mathbf{x}_t^* + \mathbf{c}_t] = 0$, and the inequality (b) comes again from the optimality condition for minimization over the original Lagrangian function; i.e., $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{x}_t - \mathbf{x}_t^*] \geq 0, \ \forall \mathbf{x}_t \in \mathcal{X}$.

Combining the above two bounds, we have

$$(\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] \leq \mathbb{E}[(\nabla \Psi_t(\hat{\mathbf{x}}_t^*) - \nabla \Psi_t(\mathbf{x}_t^*))^\top (\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*)].$$

Using the strong convexity of primal objective function Ψ_t, it holds from the property of so-termed monotonic operators that [56, Theorem 2.1.9]

$$(\nabla \Psi_t(\hat{\mathbf{x}}_t^*) - \nabla \Psi_t(\mathbf{x}_t^*))^\top (\mathbf{x}_t^* - \hat{\mathbf{x}}_t^*) \geq \sigma \|\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*\|^2$$

and it leads to

$$\begin{aligned} 0 &\leq (\hat{\boldsymbol{\lambda}}^* - \boldsymbol{\lambda}^*)^\top \mathbb{E}[\mathbf{A}\hat{\mathbf{x}}_t^* + \mathbf{c}_t] - \frac{\epsilon}{2}\|\hat{\boldsymbol{\lambda}}^*\|^2 + \frac{\epsilon}{2}\|\boldsymbol{\lambda}^*\|^2 \\ &\leq -\sigma\mathbb{E}[\|\hat{\mathbf{x}}_t^* - \mathbf{x}_t^*\|^2] - \frac{\epsilon}{2}\|\hat{\boldsymbol{\lambda}}^*\|^2 + \frac{\epsilon}{2}\|\boldsymbol{\lambda}^*\|^2 \end{aligned}$$

thus completing the proof.

5.5.2 Proof of Lemma 12

In order to arrive the statement in Lemma 12, we first introduce a helpful lemma from [23, Theorem 3] to relate the sub-optimality on a subset $\hat{\mathcal{S}}$ to the sub-optimality bound on set $\mathcal{S}$ where $\hat{\mathcal{S}} \subseteq \mathcal{S}$.

Lemma 14. *Let m and n denote the sample sizes of set $\hat{\mathcal{S}}$ and $\mathcal{S}$, i.e., $m := |\hat{\mathcal{S}}|,\ n = |\mathcal{S}|$, $m < n$, and $\boldsymbol{\lambda}_{\hat{\mathcal{S}}}$ an ω-optimal solution for the training subset $\hat{\mathcal{S}}$, i.e., $\mathbb{E}[\hat{\mathcal{D}}_{\hat{\mathcal{S}}}^* - \hat{\mathcal{D}}_{\hat{\mathcal{S}}}(\boldsymbol{\lambda}_{\hat{\mathcal{S}}})] \leq \omega$. Then the sub-optimality of $\boldsymbol{\lambda}_{\hat{\mathcal{S}}}$ for the set $\mathcal{S}$ is bounded w.h.p. in the choice of $\hat{\mathcal{S}}$ as*

$$\mathbb{E}_{\mathcal{S}}[\hat{\mathcal{D}}_{\mathcal{S}}^* - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_{\hat{\mathcal{S}}})] \leq \omega + \frac{n-m}{n}\mathcal{H}_s(m). \tag{5.41}$$

This lemma indicates that if we adaptively increase the training set from $\hat{\mathcal{S}}$ to $\mathcal{S}$, then we can bound the optimization error on a larger set $\mathcal{S}$ by the original error ω plus an additional "switching cost" $(n-m)\mathcal{H}_s(m)/n$.

Next we further define an upper bound of the optimization error $\mathcal{U}_o(k;n)$ for incrementally running offline SAGA k iterations (including both offline phase and online phase) in a training set $\mathcal{S}$ with the current sample size n; i.e., $\mathbb{E}[\hat{\mathcal{D}}_{\mathcal{S}}^* - \hat{\mathcal{D}}_{\mathcal{S}}(\boldsymbol{\lambda}_k)] \leq \mathcal{U}_o(k;n)$. Here we use index k to differentiate the real-time slot t, and we have the relation that $k = KN_t$ under online SAGA. Based on Lemma 14 and the convergence rate in Theorem 3, we can have the recursion for $\mathcal{U}_o(k;n)$ as

$$\mathcal{U}_o(k;n) = \min \begin{cases} \Gamma_n \mathcal{U}_o(k-1;n) \\ \min\limits_{m<n} \left[\mathcal{U}_o(k;m) + \dfrac{n-m}{n}\mathcal{H}_s(m)\right] \end{cases} \tag{5.42}$$

where the initial upper bound $\mathcal{U}_o(0;n) = \xi,\ \forall n$ defined in Lemma 12. We refer interested readers to see the proof of [23, Lemma 8] for which the initial upper bound holds w.h.p.

Now we are ready to prove Lemma 14 using the induction. Starting from the base case $N_0 = N^{\text{off}} = \kappa$, it follows from

$$\mathcal{U}_o(KN^{\text{off}};N_t) \leq (1-1/N_t)^{KN_t}\mathcal{U}_o(0;N_t) \overset{(a)}{\leq} \frac{\xi}{e^K} \tag{5.43}$$

where the inequality (a) is due to $(1-1/N_t)^{N_t} \leq 1/e$. Assuming the result (5.31) holds for slot t with N_t, and we have [cf. $N_{t+1} = N_t + 1$]

$$\begin{aligned}
\mathcal{U}_o(KN_{t+1}; N_{t+1}) &\overset{(b)}{\leq} \Gamma^K_{N_{t+1}} \left[\mathcal{U}_o(KN_t; N_t) + \frac{1}{N_{t+1}}\mathcal{H}_s(N_t)\right] \\
&\overset{(c)}{\leq} \left(\frac{N_t}{N_{t+1}}\right)^K \left[c\mathcal{H}_s(N_t) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_t}\right)^K + \frac{1}{N_{t+1}}\mathcal{H}_s(N_t)\right] \\
&\leq \left(\frac{N_t}{N_{t+1}}\right)^K \left(c + \frac{1}{N_{t+1}}\right)\mathcal{H}_s(N_t) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_{t+1}}\right)^K \\
&\overset{(d)}{\leq} \left(\frac{N_t}{N_{t+1}}\right)^{K-1} \left(c + \frac{1}{N_{t+1}}\right)\mathcal{H}_s(N_{t+1}) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_{t+1}}\right)^K \\
&\overset{(e)}{\leq} c\mathcal{H}_s(N_{t+1}) + \frac{\xi}{e^K}\left(\frac{\kappa}{N_{t+1}}\right)^K
\end{aligned}$$

where the inequality (b) uses the recursion (5.42) twice; the inequality (c) is due to the assumption that the argument holds for N_t; the inequality (d) uses the fact

$$\frac{N_t}{N_{t+1}} \cdot \frac{\mathcal{H}_s(N_t)}{\mathcal{H}_s(N_{t+1})} = \left(\frac{N_t}{N_{t+1}}\right)^{1-\beta} \leq 1,\ 0 \leq \beta \leq 1$$

and the inequality (e) comes from the condition (5.32). The proof of Lemma 12 thus completes.

5.5.3 Proof of Lemma 13

The following results are closely related to the *locally smooth* structure of the dual function [38], but here we use the concept of the strong concavity of the dual function to establish the arguments.

Since $\boldsymbol{\lambda}_t$ converges to $\boldsymbol{\lambda}^*,\ w.p.1$, and $\boldsymbol{\theta} > 0$, there always exists a finite time T_δ such that for $t > T_\delta$, we have $\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}_t\| \leq \theta$ and thus $\tilde{\boldsymbol{\theta}}_t \geq 0$ according to the definition of $\tilde{\boldsymbol{\theta}}_t$. Hence, we have

$$\begin{aligned}
\|\mathbf{Q}_{t+1} - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 &= \|[\mathbf{Q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{c}_t]^+ - [\tilde{\boldsymbol{\theta}}_t/\mu]^+\|^2 \\
&\overset{(a)}{\leq} \|\mathbf{Q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{c}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 \\
&\overset{(b)}{\leq} \|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 + M + 2(\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu)^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) \\
&= \|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 + M + 2\left(\frac{\boldsymbol{\gamma}_t - \boldsymbol{\lambda}^*}{\mu}\right)^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) \\
&\overset{(c)}{\leq} \|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 + \frac{2}{\mu}\left(\mathcal{D}_t(\boldsymbol{\gamma}_t) - \mathcal{D}_t(\boldsymbol{\lambda}^*)\right) + M
\end{aligned} \tag{5.44}$$

where the inequality (a) uses the non-expansive property of the projection operator; the inequality (b) uses the upper bound $M := \max_t \max_{\mathbf{x}_t \in \mathcal{X}} \|\mathbf{A}\mathbf{x}_t + \mathbf{c}_t\|^2$; and the inequality (c) follows from the fact that $\mathbf{A}\mathbf{x}_t + \mathbf{c}_t$ is a subgradient of the dual function $\mathcal{D}_t(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda} = \boldsymbol{\gamma}_t$ [cf. (5.28)].

Using the strong concavity of the dual function $\mathcal{D}(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, if follows that

$$\begin{aligned}\mathcal{D}(\boldsymbol{\gamma}_t) \leq & \mathcal{D}(\boldsymbol{\lambda}^*) + \nabla \mathcal{D}(\boldsymbol{\lambda}^*)^\top (\boldsymbol{\gamma}_t - \boldsymbol{\lambda}^*) - \frac{\epsilon}{2}\|\boldsymbol{\gamma}_t - \boldsymbol{\lambda}^*\|^2 \\ \overset{(d)}{\leq} & \mathcal{D}(\boldsymbol{\lambda}^*) - \frac{\epsilon}{2}\|\boldsymbol{\gamma}_t - \boldsymbol{\lambda}^*\|^2 \end{aligned} \tag{5.45}$$

where the inequality (d) follows from the optimality condition of the dual problem $\nabla \mathcal{D}(\boldsymbol{\lambda}^*)^\top (\boldsymbol{\gamma}_t - \boldsymbol{\lambda}^*) \leq 0$.

Then using (5.45), and taking expectations on (5.44) over the random state $\mathbf{s}_t$ conditioning on $\mathbf{Q}_t$, we have

$$\mathbb{E}\left[\|\mathbf{Q}_{t+1} - \tilde{\boldsymbol{\theta}}_t/\mu\|^2\right] \overset{(c)}{\leq} \|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 - \mu\epsilon\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 + M \tag{5.46}$$

where we use the fact that $\mathcal{D}(\boldsymbol{\lambda}) = \mathbb{E}\left[\mathcal{D}_t(\boldsymbol{\lambda})\right]$.

Hence, based on (5.46), if we have

$$-\mu\epsilon\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\|^2 + M \leq -2\sqrt{\mu}\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\| + \mu \tag{5.47}$$

then it follows that

$$\mathbb{E}\left[\|\mathbf{Q}_{t+1} - \tilde{\boldsymbol{\theta}}_t/\mu\|^2\right] \leq \left(\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\| - \sqrt{\mu}\right)^2 \tag{5.48}$$

which implies the argument (5.37) in the lemma. By checking the Vieta's formulas for second-order equations, there exists $B = \Theta(\frac{1}{\sqrt{\mu}})$ such that for $\|\mathbf{Q}_t - \tilde{\boldsymbol{\theta}}_t/\mu\| > B$, the inequality (5.47) holds, so the lemma readily follows.

5.5.4 Proof of Theorem 5

Note that Lemma 13 assures that the queue length $\mathbf{Q}_t$ always tracks a time-varying target $\tilde{\boldsymbol{\theta}}_t/\mu$. However, as $\lim_{t\to\infty} \tilde{\boldsymbol{\theta}}_t/\mu = \boldsymbol{\theta}/\mu, \ w.p.1$, the queue length $\mathbf{Q}_t$ will eventually track $\boldsymbol{\theta}/\mu$ and deviate within the distance of $B = \Theta(\frac{1}{\sqrt{\mu}})$, which implies that a steady-state of the Markov chain $\{\mathbf{Q}_t\}$ exists according to the Foster-Lyapunov Theorem [51]. A rigorous proof can follow the lines of that in [37, Theorem 1], and we omit here due to limited space.

5.5.5 Proof of Theorem 6

Intuitively, the effective dual variable $\lim_{t\to\infty}\boldsymbol{\gamma}_t = \boldsymbol{\lambda}^* + \mu\mathbf{Q}_t - \boldsymbol{\theta},\ w.p.1$, the proposed online SAGA will behave always the same as the FQLA-ideal algorithm of [38] in the steady state, thus the asymptotical performance follows from the FQLA-ideal algorithm. For completeness, the detailed proof follows.

Defining the Lyapunov drift as $\Delta(\mathbf{Q}_t) := \frac{1}{2}(\|\mathbf{Q}_{t+1}\|^2 - \|\mathbf{Q}_t\|^2)$, and squaring the queue update, we have

$$\begin{aligned}\|\mathbf{Q}_{t+1}\|^2 =& \|\mathbf{Q}_t\|^2 + 2\mathbf{Q}_t^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) + \|\mathbf{A}\mathbf{x}_t + \mathbf{c}_t\|^2 \\ \overset{(a)}{\leq}& \|\mathbf{Q}_t\|^2 + 2\mathbf{Q}_t^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) + M\end{aligned}$$

where the inequality (a) follows from the upper bound of $\|\mathbf{A}\mathbf{x}_t + \mathbf{c}_t\|^2$. Multiplying $\mu/2$ on each side and adding $\Psi_t(\mathbf{x}_t)$, it yields

$$\begin{aligned}\mu\Delta(\mathbf{Q}_t) + \Psi_t(\mathbf{x}_t) =& \Psi_t(\mathbf{x}_t) + \mu\mathbf{Q}_t^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) + \frac{\mu M}{2} \\ \overset{(b)}{=}& \Psi_t(\mathbf{x}_t) + (\boldsymbol{\gamma}_t - \boldsymbol{\lambda}_t + \boldsymbol{\theta})^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) + \frac{\mu M}{2} \\ \overset{(c)}{=}& \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\gamma}_t) + (\boldsymbol{\theta} - \boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t) + \frac{\mu M}{2}\end{aligned}$$

where the equality (b) uses the definition of $\boldsymbol{\gamma}_t$ and the equality (c) is the definition of the instantaneous Lagrangian. Taking expectations over the random state $\mathbf{s}_t$ conditioning on $\mathbf{Q}_t$, we have

$$\begin{aligned}&\mu\mathbb{E}\left[\Delta(\mathbf{Q}_t)\right] + \mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right] \\ \overset{(d)}{=}& \mathcal{D}(\boldsymbol{\gamma}_t) + \mathbb{E}\left[(\boldsymbol{\theta} - \boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t)\right] + \frac{\mu M}{2} \\ \overset{(e)}{\leq}& \Psi^* + \mathbb{E}\left[(\boldsymbol{\theta} - \boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t + \mathbf{c}_t)\right] + \frac{\mu M}{2}\end{aligned}$$

where the equality (d) follows from the definition of the dual function (5.12), while the inequality (e) uses the weak duality and the fact that $\tilde{\Psi}^* \leq \Psi^*$.

Summing both sides over $t = 0, \ldots, T-1$, dividing both sides by T and letting $T \to \infty$, we

arrive at

$$\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\Psi_t(\mathbf{x}_t)\right]\\
\overset{(f)}{\leq}&\Psi^*+\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big[(\boldsymbol{\theta}-\boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t+\mathbf{c}_t)\Big]+\frac{\mu M}{2}+\lim_{T\to\infty}\frac{\mu\|\mathbf{Q}_0\|^2}{2T}\\
\leq&\Psi^*+\frac{\mu M}{2}+\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(\boldsymbol{\theta}-\boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t+\mathbf{c}_t)\right]
\end{aligned}\tag{5.49}$$

where the inequality (f) comes from $\|\mathbf{Q}_{T+1}\|^2\geq 0$.

Now it remains to show that the last term in RHS of (5.49) is $\mathcal{O}(\mu)$. Since $\boldsymbol{\lambda}_t$ converges to $\boldsymbol{\lambda}^*,\ w.p.1$, there always exists a finite time T_δ such that for $t>T_\delta$, we have $\|\boldsymbol{\lambda}^*-\boldsymbol{\theta}-(\boldsymbol{\lambda}_t-\boldsymbol{\theta})\|\leq\delta,\ w.p.1$. Hence, we have

$$\begin{aligned}
&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(\boldsymbol{\theta}-\boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t+\mathbf{c}_t)\right]\\
\overset{(g)}{\leq}&\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(\boldsymbol{\lambda}^*-\boldsymbol{\theta})^\top(-\mathbf{c}_t-\mathbf{A}\mathbf{x}_t)\right]+\mathcal{O}(\delta)
\end{aligned}\tag{5.50}$$

where the inequality (g) follows from $T_\delta<\infty$ and $\|\mathbf{A}\mathbf{x}_t+\mathbf{c}_t\|$ is bounded. With q_i denoting the ith entry of the vector $-\mathbf{c}_t-\mathbf{A}\mathbf{x}_t$, by the large deviation bound in [37, Lemma 4] and [38, Theorem 4], there exist constant $D_1=\Theta(\frac{1}{\mu})$, $D_2=\Theta(\sqrt{\mu})$ and $\tilde{B}=\Theta(\frac{1}{\sqrt{\mu}})$ that

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[q_i]\leq\sqrt{M}D_1e^{-D_2(\theta_i/\mu-\tilde{B}-\sqrt{M})},\ w.p.1.\tag{5.51}$$

By setting $\theta_j=\sqrt{\mu}\log^2(\mu)$, there exists a sufficiently small μ such that $-D_2(\frac{1}{\sqrt{\mu}}\log^2(\mu)-\tilde{B}-\sqrt{M})\leq 2\log(\mu)$, then it follows that $\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[q_i]\leq\sqrt{M}D_1\mu^2=\mathcal{O}(\mu)$. Setting $\delta=\mathcal{O}(\mu)$, we have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(\boldsymbol{\theta}-\boldsymbol{\lambda}_t)^\top(\mathbf{A}\mathbf{x}_t+\mathbf{c}_t)\right]=\mathcal{O}(\mu)\tag{5.52}$$

from which the proof completes.

Chapter 6

Conclusions and Future Works

6.1 Summarizing Conclusions

The present thesis dealt with developing interdisciplinary approaches to offer a sustainable solution for resource allocation in future cloud networks, by leveraging advances in the areas of smart grids, big data, and machine learning.

Targeting this goal, we introduced elegant mathematical models, optimization frameworks, and developed low-complexity algorithms to account for three different aspects: stochasticity of resources (Chapters 2 and 3), system robustness (Chapter 4), and big data-driven learning approaches (Chapter 5).

The thesis started with stochastic resource allocation tailored for a single data center in Chapter 2. By pointing out that a substantial amount of energy in data centers is consumed by the cooling infrastructure, the thesis introduced a practical cooling system composed of two cooling approaches with different cooling coefficients, along with which workload operating systems, and DC power supply system were put forward. Building upon these models, the resource allocation task was formulated as an infinite time horizon optimization problem aiming to minimize the limiting time-average operational net-cost, where the challenge is to come up with online decisions under time-coupling constraints with only causal information. To tackle this issue, the decision variables were decoupled across time. Then, leveraging Lagrange relaxation and stochastic approximation techniques, a novel online control algorithm was proposed to allocate resources "on the

fly." Based on the revealed characteristics of the optimal schedules, it was formally established that the proposed algorithm yields a feasible and near-optimal resource management strategy for the original problem. Numerical tests further demonstrated that the proposed algorithm works well not only for i.i.d. processes, but also in real-data scenarios, where the underlying randomness is highly correlated over time.

While the resource allocation task in Chapter 2 presumed that data requests are processed locally or optimal routing has been performed, Chapter 3 considered the joint workload and energy management for a cloud network consisting of multiple geo-distributed MNs and DCs. The proposed workload routing and scheduling policy of Chapter 3 includes both DWs and IWs, and the energy management scheme integrates the RES, storage units and two-way energy trading, to minimize the total energy cost from cooling and IT operating systems. Furthermore, leveraging the flexibility provided by demand-response programs, an incentive payment mechanism was introduced to modulate the peaks of the IW demand, while guaranteeing user quality of services. Targeting a space-time decoupled online solver, this thesis developed a novel two-timescale algorithm was also developed in which the Lagrange multipliers of the time-coupling constraints were updated using a stochastic approximation technique at a slow timescale, while the multipliers of the MN-DC coupling constraints were updated using FISTA at a fast timescale. The novel algorithm incurs low communication overhead, and can be implemented in a distributed fashion. If the random process involved is either independent and identically distributed or follows a finite-state ergodic Markov chain, it is further established that the proposed schemes yield a feasible and asymptotically optimal resource-management strategy for the original problem.

Approaches in Chapters 2 and 3 dealing with RES uncertainty mainly rely on the stochastic approximation technique. To guarantee convergence and optimality, these methods typically assume independent and identically distributed RES samples. Chapter 4 considered the robust workload and energy management for a cloud network. Distinct from Chapters 2 and 3 and many existing works, a deterministic uncertainty set of the unknown renewable generation was introduced to account for the stochastic and nondispatchable nature of RES. The proposed uncertainty set of the RES generation only requires easy-to-obtain first-order statistics, and possibly sample correlation statistics. Control parameters are further designed to trade off robustness for conservatism of the robust opti-

mization formulation. Adopting again practical models, the resource allocation task was formulated in Chapter 4 as a robust optimization problem, which minimizes the system's worst-case net cost subject to DCs' operational constraints. Leveraging the problem structure, this problem was cast as a convex program. Capitalizing on the dual decomposition approach, an efficient distributed solver was developed. It was shown that the proposed algorithm is guaranteed to yield the desired robust workload and energy-management strategy, and could also facilitate distributed implementations among the MNs and DCs. Finally, extensive numerical tests with real data corroborated the merits of the proposed framework and approaches.

The main limitation of existing stochastic resource allocation schemes in Chapters 2 and 3 is the relatively slow convergence rate and the high network delay as a by-product. To cope with this, Chapter 5 developed a comprehensive learning protocol to integrate statistical learning techniques into the stochastic resource allocation tasks that are tackled using nonlinear optimization tools. The considered stochastic resource allocation problem in Chapters 2 and 3 was formulated as a machine learning task with the goal of learning the Lagrange multipliers in a fast and efficient manner. Casting the batch learning problem in the form of maximizing the sum of finite concave functions, we connected it with a prevalent machine learning routine termed ERM. Capitalizing on this problem structure, the so-called SAGA approach was adopted for the problem at hand. It was shown in Chapter 5 that the offline SAGA can efficiently compute an empirical Lagrange multiplier with an order-optimal convergence rate at a fairly low computational cost per iteration. In the online setting, a novel online SAGA scheme was developed in Chapter 5 to operate in a learning-while-testing fashion. Online SAGA can be viewed as a combination of stochastic approximation and statistical learning: in the learning phase, it preserves the simple update of offline SAGA to dynamically learn from streaming data and maintain the learning error always below the statistical accuracy; while in the testing phase, it incorporates merits of well-studied stochastic dual subgradient to explicitly track the queue variations and guarantee the long-term queue stability. Finally, it has been established - both theoretically and empirically - that online SAGA can improve the delay and convergence performance of existing resource allocation schemes, at the cost of only two more sample evaluations per time slot.

6.2 Future Directions

The results in this thesis open up a number of future research directions. In a nutshell, there is ample room for improving online resource allocation approaches proposed in two different but complementary venues: exploring the value of *historical data*, and leveraging the (possibly imperfect) *prediction* of future system states.

Chapter 5 of the thesis serves as an exciting first step towards integrating statistical learning techniques into the stochastic resource allocation tasks. The performance gain – from both theoretical and numerical merits – highlights the potential of learning from historical data. However, there are several challenges that prompt further research efforts. For instance, the number of historical data needed will monotonically increase with time. Hence, it is of interest to derive means of selecting an informative subset of big historical data for learning. From the learning strategy perspective, the proposed approach in Chapter 5 belongs to the class of first-order algorithms, which opens up the possibility of leveraging second-order information in an efficient and appropriate manner.

On the other hand, most existing online algorithms including those introduced in this thesis assume that future system information is either stochastic (most cases i.i.d.) or adversarial, and do not explore the virtues of predicted information in online resource allocation, or, in general online convex optimization. However, there is no doubt that predictions, though not accurate, are usually available in practice, and recent progress in machine learning further makes such predictions increasingly reliable. Therefore, postulating prediction models for future information to balance the tradeoff of stochasticity and adversity are worthwhile subjects deserving further investigation.

Bibliography

[1] "Apple environmental responsibility." [Online]. Available: http://www.apple.com/environment/

[2] "CAISO hourly renewables watch." [Online]. Available: http://www.caiso.com/green/renewableswatch.html

[3] "Google data center locations." [Online]. Available: http://www.google.com/about/datacenters/

[4] "MISO market data." [Online]. Available: https://www.midwestiso.org/MarketsOperations/RealTimeMarketData/

[5] "CVX: Matlab software for disciplined convex programming, version 2.1," Sep. 2012. [Online]. Available: http://cvxr.com/cvx

[6] "California ISO report for wind/solar resources 2012," Jun. 2013. [Online]. Available: http://www.caiso.com/Documents/2012Report-Wind-SolarResourcesPostedJun5_2013.htm

[7] "Hourly electric supply charges in New York," Jan. 2015. [Online]. Available: https://www.nationalgridus.com/

[8] "Pennsylvania-New Jersey-Maryland interconnection (PJM) hourly real-time wind generation," Jan. 2015. [Online]. Available: http://www.pjm.com/markets-and-operations/ops-analysis.aspx

[9] 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/

[10] S. Asmussen, *Applied Probability and Queues*. Berlin, Germany: Springer Science & Business Media, 2008, vol. 51.

[11] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," in *Proc. ACM SIGMETRICS*, vol. 30, no. 1, Jun. 2002, pp. 195–205.

[12] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[13] A. Beck, A. Nedic, A. Ozdaglar, and M. Teboulle, "An $\mathcal{O}(1/k)$ gradient method for network resource allocation problems," vol. 1, no. 1, pp. 64–73, Mar. 2014.

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

[15] D. P. Bertsekas, *Convex Optimization Theory*. Belmont, MA: Athena Scientific, 2009.

[16] ——, *Nonlinear programming*. Belmont, MA: Athena scientific, 1999.

[17] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," vol. 28, no. 1, pp. 52–63, Feb. 2013.

[18] C. M. Bishop, "Pattern recognition," 2006.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY: Cambridge University Press, 2004.

[20] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *Lecture notes of EE392o, Stanford University, Autumn Quarter*, 2003.

[21] T. Chen, X. Wang, and G. B. Giannakis, "Cooling-aware energy and workload management in data centers via stochastic optimization," to appear 2016.

[22] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *Proc. IEEE MASCOTS*, Singapore, Jul. 2011, pp. 390–399.

[23] H. Daneshmand, A. Lucchi, and T. Hofmann, "Starting small - learning with adaptive sample sizes," in *Proc. Intl. Conf. on Machine Learning*, New York, NJ, Jun. 2016. [Online]. Available: http://arxiv.org/abs/1603.02839

[24] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Info. Process. Syst.*, Montral, Canada, Dec. 2014, pp. 1646–1654.

[25] W. Deng, F. Liu, H. Jin, C. Wu, and X. Liu, "Multigreen: Cost-minimizing multi-source datacenter power supply with online control," in *Proc. ACM Intl. Conf. on Future Energy systems*, Berkeley, CA, May 2013, pp. 149–160.

[26] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and mac for stability and fairness in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[27] Federal Energy Regulatory Commission, *Energy Primer: a Handbook of Energy Market Basics*, Washington, DC, 2015.

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

[29] N. Gatsis and A. G. Marques, "A stochastic approximation approach to load shedding inpower networks," Florence, Italy, May 2014.

[30] N. Gatsis, A. Ribeiro, and G. B. Giannakis, "A class of convergent algorithms for resource allocation in wireless fading networks," vol. 9, no. 5, pp. 1808–1823, May 2010.

[31] A. M. González, A. M. S. Roque, and J. García-González, "Modeling and forecasting electricity prices with input/output hidden Markov models," vol. 20, no. 1, pp. 13–24, Feb. 2005.

[32] Y. Guo and Y. Fang, "Electricity cost saving strategy in data centers by using energy storage," vol. 24, no. 6, pp. 1149–1160, Jun. 2013.

[33] 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," vol. 25, no. 8, pp. 2030–2042, Aug. 2014.

[34] P. Harsha and M. Dahleh, "Optimal management and sizing of energy storage under dynamic pricing for the efficient integration of renewable energy," vol. 30, no. 3, pp. 1164–1181, May 2015.

[35] Y. He, S. Elnikety, J. Larus, and C. Yan, "Zeta: Scheduling interactive services with partial execution," in *Proc. ACM Symposium on Cloud Computing*, San Jose, CA, Oct. 2012.

[36] J. P. Holman, *Heat Transfer*, 8th ed. Columbus, OH: McGraw-Hill, 1996.

[37] L. Huang, X. Liu, and X. Hao, "The power of online learning in stochastic network optimization," in *Proc. ACM SIGMETRICS*, vol. 42, no. 1, New York, NY, Jun. 2014, pp. 153–165.

[38] L. Huang and M. J. Neely, "Delay reduction via lagrange multipliers in stochastic network optimization," vol. 56, no. 4, pp. 842–857, Apr. 2011.

[39] A. Kolmogorov, *Foundations of the Theory of Probability.*

[40] V. Kong and X. Solo, *Adaptive Signal Processing Algorithms*. Upper Saddle River, NJ: Prentice Hall, 1995.

[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. Lakshminaryana, H. V. Poor, and T. Quek, "Cooperation and storage trade-offs in power grids with renewable energy resources," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1–12, Jul. 2014.

[43] R. H. Lasseter and P. Paigi, "Microgrid: A conceptual solution," in *Proc. of IEEE Power Electronics Specialists Conf.*, Aachen, Germany, Jun. 2004, pp. 4285–4290.

[44] B. Li, R. Li, and A. Eryilmaz, "On the optimal convergence speed of wireless scheduling for fair resource allocation," *Networking, IEEE/ACM Transactions on*, vol. 23, no. 2, pp. 631–643, Apr. 2015.

[45] S. Li, M. Brocanelli, W. Zhang, and X. Wang, "Integrated power management of data centers and electric vehicles for energy and regulation market participation," vol. 5, no. 5, pp. 2283–2294, Sep. 2014.

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

[47] J. Liu, A. Eryilmaz, N. B. Shroff, and E. S. Bentley, "Heavy-ball: A new approach to tame delay and convergence in wireless network optimization," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2016.

[48] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. ACM SIGMETRICS*, vol. 40, no. 1, London, UK, Jun. 2012, pp. 175–186.

[49] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing data center demand response," *Elsevier Performance Evaluation*, vol. 42, no. 1, pp. 111–123, 2014.

[50] 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," vol. 61, no. 6, pp. 2789–2807, Jul. 2012.

[51] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Berlin, Germany: Springer Science & Business Media, 2012.

[52] A. Mokhtari and A. Ribeiro, "Adaptive newton method for empirical risk minimization to statistical accuracy," *arXiv preprint, arXiv:1605.07659*, May 2016.

[53] A. Nedic and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM J. Optimization*, vol. 19, no. 4, pp. 1757–1780, 2009.

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

[55] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[56] ——, *Introductory lectures on convex optimization: A basic course*. Berlin, Germany: Springer, 2013, vol. 87.

[57] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[58] A. Papavasiliou and S. S. Oren, "Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network," *Operations Research*, vol. 61, no. 3, pp. 578–592, May 2013.

[59] C. Patel, R. Sharma, C. Bash, and A. Beitelmal, "Energy flow in the information technology stack," in *Proc. IMECE*, Chicago, IL, Nov. 2006.

[60] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *Journal of Nonlinear and Convex Analysis*, to appear 2014.

[61] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," in *Proc. ACM SIGCOMM*, vol. 39, no. 4, Barcelona, Spain, Aug. 2009, pp. 123–134.

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

[63] D. Rastler, "Electricity energy storage technology options," *White Paper*, 2010. [Online]. Available: www.epri.com

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

[65] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[66] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence rate for finite training sets," in *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2012, pp. 2663–2671.

[67] E. Samadiani, Y. Joshi, and F. Mistree, "The thermal design of a next generation data center: a conceptual exposition," in *Proc. IEEE Intl. Conf. on Thermal Issues in Emerging Tech.: Theory and Application*, Cairo, Egypt, Jan. 2007, pp. 93–102.

[68] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia, PA: SIAM, 2009.

[69] V. Solo, "Averaging analysis of adaptive algorithms made simple," in *System Identification, Environmental Modelling, and Control System Design*. Springer-Verlag, London, 2012, pp. 115–131.

[70] S. Sun, M. Dong, and B. Liang, "Joint supply, demand, and energy storage management towards microgrid cost minimization," Venice, Italy, Nov. 2014.

[71] K. C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3 — a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 2009.

[72] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proc. ACM SIGMETRICS*, San Jose, CA, Jun. 2011, pp. 221–232.

[73] V. Valls and D. J. Leith, "Descent with approximate multipliers is enough: Generalising max-weight," *arXiv preprint, arXiv:1511.02517*, Nov. 2015.

[74] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996.

[75] V. Vapnik, *The nature of statistical learning theory*. Berlin, Germany: Springer Science & Business Media, 2013.

[76] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: Decentralized server selection for cloud services," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 231–242, 2011.

[77] J. Whitney and P. Delforge, "Data center efficiency assessment," *Issue Paper*, 2015. [Online]. Available: http://www.nrdc.org/energy/data-center-efficiency-assessment.asp

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

[79] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York, NY: Wiley & Sons, 1996.

[80] ——, *Power Generation, Operation, and Control*. Hoboken, NJ: John Wiley & Sons, 2012.

[81] D. Xu and X. Liu, "Geographic through filling for Internet datacenters," in *Proc. IEEE INFOCOM*, Orlando, FL, Mar. 2012, pp. 2881–2885.

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

[83] ——, "Reducing electricity demand charge for data centers with partial execution," in *Proc. Intl. Conf. on Future Energy Systems*, Cambridge, UK, Jun. 2014, pp. 51–61.

[84] E. Yao, P. Samadi, V. Wong, and R. Schober, "Residential demand side management under high penetration of rooftop photovoltaic units," to appear 2015.

[85] 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*, Orlando, FL, Mar. 2012, pp. 1431–1439.

[86] M. Zargham, A. Ribeiro, and A. Jadbabaie, "Accelerated backpressure algorithm," *arXiv preprint, arXiv:1302.1475*, Feb. 2013.

[87] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network flow optimization," vol. 59, no. 4, pp. 905–920, Apr. 2014.

[88] 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*, Kowloon, Hong Kong, Apr. 2015, pp. 2632–2640.

[89] Y. Zhang, N. Gatsis, and G. B. Giannakis, "Robust energy management for microgrids with high-penetration renewables," vol. 4, no. 4, pp. 944–953, Oct. 2013.

[90] Y. Zhang, S.-J. Kim, and G. B. Giannakis, "Short-term wind power forecasting using nonnegative sparse coding," in *Proc. of IEEE Conf. on Info. Sci. and Syst.*, Baltimore, MD, Mar. 2015, pp. 1–5.

[91] R. Zhou, Z. Wang, A. McReynolds, C. E. Bash, T. W. Christian, and R. Shih, "Optimization and control of cooling microgrids for data centers," in *Proc. IEEE Conf. on Thermal and Thermomech. Phenomena in Electron. Syst.*, San Diego, CA, May 2012, pp. 338–343.