A Virtual-Queue-Based Algorithm for Constrained Online Convex Optimization With Applications to Data Center Resource Allocation

Xuanyu Cao, Junshan Zhang, *Fellow, IEEE*, and H. Vincent Poor, *Fellow, IEEE*

***Abstract*—In this paper, online convex optimization (OCO) problems with time-varying objective and constraint functions are studied from the perspective of an agent who takes actions in real time. Information about the current objective and constraint functions is revealed only after the corresponding action is already chosen. Inspired by a fast converging algorithm for time-invariant optimization in the very recent work [1], we develop a novel online algorithm based on virtual queues for constrained OCO. Optimal points of the dynamic optimization problems with full knowledge of the current objective and constraint functions are used as a dynamic benchmark sequence. Upper bounds on the regrets with respect to the dynamic benchmark and the constraint violations are derived for the presented algorithm in terms of the temporal variations of the underlying dynamic optimization problems. It is observed that the proposed algorithm possesses sublinear regret and sublinear constraint violations, as long as the temporal variations of the optimization problems are sublinear, i.e., the objective and constraint functions do not vary too drastically across time. The performance bounds of the proposed algorithm are superior to those of the state-of-the-art OCO method in most scenarios. Besides, different from the saddle point methods widely used in constrained OCO, the stepsize of the proposed algorithm does not rely on the total time horizon, which may be unknown in practice. Finally, the algorithm is applied to a dynamic resource allocation problem in data center networks. Numerical experiments are conducted to corroborate the merit of the developed algorithm and its advantage over the state-of-the-art.**

***Index Terms*—Online convex optimization, constrained optimization, sequential decision making, virtual queues, dynamic resource allocation, data centers.**

I. Introduction

In the past decade, online convex optimization (OCO) has emerged as a promising paradigm for tackling many signal processing and resource allocation issues involving time-variability and uncertainty [2], [3]. Unlike traditional time-invariant optimization [4], [5], OCO can be viewed as a sequential decision making procedure of an agent, who decides an action at each time. The time-varying objective function and/or constraint functions are unknown to the agent a priori. Further, feedback information of the objective and constraint functions is revealed to the agent only after the action of the agent is already determined. Because of this lack of just-in-time information, it is impossible for OCO algorithms to find the exact optimal point in each time. Instead, OCO seeks to minimize the *regret*, i.e., the performance gap between the actions induced by the algorithm and some benchmark actions (e.g., the offline optima in hindsight). Clearly, a sublinear regret is desirable since it implies that the time average performance of the algorithm is no worse than that of the benchmark asymptotically. Such an OCO framework arises in many applications in which the environment is dynamic and uncertain, e.g., smart grids with intermittent supply of renewable energy [6], [7] and data centers with dynamic user demands [8]–[10].

Owing to its broad applicability, various forms of OCO problems have been investigated extensively in the recent decade. In the seminal work [11], Zinkevich initiated the study of unconstrained OCO and proposed an online gradient descent algorithm with sublinear regret of[1] $\mathcal{O}\left(\sqrt{T}\right)$ (T is the total time horizon) compared to the static offline optima. The regret was further reduced to $\mathcal{O}(\log T)$ by several online algorithms presented in [12]. In [11], regret bound with respect to dynamic benchmark was also presented, in which the "path length" (or temporal variation) of the benchmark sequence was bounded. Since dynamic benchmarks were more meaningful in many applications, they were adopted in [13] and [14], in which algorithms with sublinear regrets compared to dynamic benchmarks were developed. Moreover, Flaxman *et al.* examined OCO with bandit feedback, in which only the values of the objective functions at the chosen actions (instead of the entire functions) were revealed [15]. A modified online gradient method with $\mathcal{O}\left(T^{\frac{3}{4}}\right)$ regret was proposed in [15] based on single-point estimates of gradients since only one-point information about the objective function is available in the bandit setting. Later, Agarwal *et al.* proposed

Manuscript received September 30, 2017; revised February 27, 2018; accepted March 26, 2018. Date of publication April 16, 2018; date of current version July 27, 2018. This work was supported by the U.S. Army Research Office under Grant W911NF-16-1-0448. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Deepa Kundur. *(Corresponding author: Xuanyu Cao.)*

X. Cao and H. Vincent Poor are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: x.cao@princeton.edu; poor@princeton.edu).

J. Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: junshan.zhang@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2018.2827302

[1]We follow the standard definition of $\mathcal{O}(\cdot)$. That is, for positive sequences a_n and b_n, $a_n = \mathcal{O}(b_n)$ if there is a positive constant c such that $a_n \le cb_n$ for sufficiently large n.

to use multi-point bandit feedback to improve the regret performance in [16]. In addition, Chen *et al.* took action switching costs and noisy predictions of the dynamic objective functions into consideration in [17], [18].

The aforementioned papers focused on unconstrained OCO, whereas many practical OCO problems involved constraints, which might be time-varying as well. This discrepancy motivated several recent works on constrained OCO. In [19], constrained OCO with time-invariant constraints and static benchmark was studied by Mahdavi *et al.* and several online variants of saddle point method were proposed for different scenarios of feedback information. Extensions to continuous-time version of constrained OCO were investigated in [20], where a saddle point type of controller was designed. Furthermore, OCO with affine equality constraints was examined in [21] by using an online version of the alternating direction method of multipliers (ADMM) [22] while distributed OCO over networks with consensus or proximity constraints were studied in [23] and [24], respectively, by invoking online versions of saddle point methods. In addition, the special case of online linear optimization problem was considered in [25] for different feedback information. The constraints of the OCO in all these works are time-invariant and known in advance. Thus, no feedback information associated with constraints is necessary. Recently, constrained OCO with time-varying constraints was studied in [26], where a dynamic benchmark was used to define the regret. Specifically, in [26], the authors proposed a modified online saddle point (MOSP) method and gave upper bounds for its regret and constraint violations in terms of the temporal variations of the dynamic optimization problems. However, the upper bounds for regret and constraint violations of MOSP were always no less than $\mathcal{O}\left(T^{\frac{2}{3}}\right)$ even when the temporal variations of the problem were very small. Besides, in order to achieve sublinear performance bounds, the stepsize parameters of MOSP should depend on the total time horizon T, which might be unknown in practice, i.e., the online optimization/learning procedure might stop at some unknown time. Even if the total time horizon T was known in advance, the performance bounds of MOSP only held for the particular time T and there was no theoretical guarantee for MOSP before time T when the online optimization procedure was still running. In fact, this reliance of stepsizes on total time horizon was common in the various versions of saddle point methods widely used for constrained OCO [19], [23], [24], [26]. Additionally, Neely and Yu developed a virtual-queue based online algorithm for constrained OCO with time-varying constraints in [27] recently. There, static offline optimum is used to define the regret, which may not be a reasonable benchmark if the underlying system is inherently time-varying and static optimum is not very meaningful.

In this paper, instead of using saddle point method and its variants, we develop an online version of the virtual queue algorithm presented in [1], which exhibits faster convergence rate than classical constrained optimization algorithms (e.g., dual gradient method) for time-invariant optimization. The stepsize parameter of the advocated algorithm does not rely on the entire time horizon and thus the online optimization procedure can terminate at any arbitrary (possibly unknown) time. Upper bounds on the regret and constraint violations of the proposed algorithm are established in terms of the temporal variations of the dynamic optimization problems. The algorithm possesses sublinear regret and constraint violations provided that the temporal variations of the dynamic problems are sublinear, i.e., the objective and constraint functions do not vary too drastically across time. The regret and constraint violation bounds of the developed algorithm are superior to those of MOSP in [26] in most scenarios, especially when the temporal variations of the dynamic problems are not overly drastic. Furthermore, we note that the performance guarantees of the proposed algorithm hold for arbitrary time instants, including those before the termination of the online optimization procedure. In contrast, the performance guarantees for saddle point methods, e.g., MOSP, generally only hold at the moment when the online procedure is ended [19], [23], [24], [26]. Finally, we apply the presented algorithm to a dynamic resource allocation problem in data center networks. In such a case, the algorithm only involves simple closed-form computation and affords distributed parallel implementation. Numerical experiments are carried out to demonstrate the merit of the algorithm and its performance gain relative to the state-of-the-art constrained OCO algorithms such as MOSP.

The organization of the rest of this paper is as follows. In Section II, we formulate the constrained OCO problem and develop an algorithm based on virtual queues. Upper bounds on the regret and constraint violations of the algorithm are derived in Section III. A case study of dynamic resource allocation in data centers is presented Section IV. In Section V, we conclude this work.

II. Problem Formulation and Virtual-Queue-Based Online Algorithm

In this section, we first formulate the constrained OCO problem with time-varying objective and constraint functions. Regrets with respect to a dynamic benchmark and constraint violations are defined as two performance criteria for online algorithms of the OCO. Then, a novel algorithm based on virtual queues is developed and its connections to existing optimization methods are discussed.

A. Problem Formulation

The classical unconstrained OCO problem can be described as the following iterative procedure between an agent and the nature [11]. Assume that time is discretized. At each time t, the agent selects an action $\boldsymbol{x}_t \in \mathbb{R}^n$ from an action set $\mathbb{X} \subset \mathbb{R}^n$. After the action $\boldsymbol{x}_t$ is chosen, the nature announces a loss function $f_t : \mathbb{R}^n \mapsto \mathbb{R}$ to the agent, who experiences a loss of $f_t(\boldsymbol{x}_t)$. The goal of the agent is to minimize this loss. The action set $\mathbb{X}$ is known in adavance to the agent before the OCO procedure starts and remains unchanged as the procedure progresses. For instance, in adaptive signal processing [28], $\boldsymbol{x}_t$ may be the estimate of the unknown parameters at time t; $f_t(\cdot)$ may correspond to some data fitting errors, e.g., the total or discounted squared errors, at time t; and $\mathbb{X}$ may be the set of possible values of the unknown parameters. As new data arrive sequentially, the fitting error function $f_t(\cdot)$ varies across time, leading to an OCO formulation.

This classical unconstrained OCO formulation, though being useful in many situations, cannot deal with problems with constraints [19], especially time-varying constraints [20], [26], which arise in many practical applications. For instance, in smart grids, with the high penetration of renewables, power supply can be uncertain and intermittent. The controller of the power grid needs to schedule the time-varying random power supply in a real time manner, which can be posed as online optimization problems with time-varying resource constraints [6], [7]. In data centers, due to the uncertain dynamic demands of users, the operators also face with online optimization with time-varying constraints. These applications motivate us to study constrained OCO with time-varying constraints in this paper. Specifically, at each time t, after an action $\boldsymbol{x}_t \in \mathbb{X}$ is chosen, the nature will reveal not only a loss function $f_t(\cdot)$ but also a vector-valued constraint function $\boldsymbol{g}_t : \mathbb{R}^n \mapsto \mathbb{R}^m$ to the agent. The agent aims at minimizing the loss $f_t(\boldsymbol{x}_t)$ while satisfying the time-varying constraints $\boldsymbol{g}_t(\boldsymbol{x}_t) \preceq \boldsymbol{0}$ (for vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^l$, $\boldsymbol{a} \preceq \boldsymbol{b}$ means $a_i \leq b_i, \forall i$), i.e., it wants to solve the following dynamic optimization problem:

$$\boldsymbol{x}_t^* \in \arg\min_{\boldsymbol{x} \in \mathbb{X}} \{f_t(\boldsymbol{x}) | \boldsymbol{g}_t(\boldsymbol{x}) \preceq \boldsymbol{0}\}. \quad (1)$$

In adaptive signal processing, the constraint function $\boldsymbol{g}_t(\cdot)$ can embody the knowledge of the unknown parameters at time t. It varies across time as the knowledge is updated when new data or measurements arrive sequentially. Nevertheless, solving problem (1) directly is impossible in the online setting here as the loss function $f_t(\cdot)$ and constraint function $\boldsymbol{g}_t(\cdot)$ are revealed only after the agent has already chosen the action $\boldsymbol{x}_t$. In particular, since $\boldsymbol{g}_t(\cdot)$ is unknown a priori, the constraint $\boldsymbol{g}_t(\boldsymbol{x}_t) \preceq \boldsymbol{0}$ is hard to be satisfied for each time t. Instead, the agent tries to satisfy the constraints in the long term. Specifically, the agent tries to enforce the long-term constraint $\sum_{t=0}^{T-1} \boldsymbol{g}_t(\boldsymbol{x}_t) \preceq \boldsymbol{0}$ over a period of T. In fact, this type of long-term constraint emerges naturally in many applications. For example, in a smart grid with renewable energy sources, the grid controller wants to balance the power demand by the the renewable energy supply. To combat the uncertainty and intermittence of renewables, the controller often reserves some traditional energy (e.g., coal and gas) to balance the possible temporary deficit of power supply from the renewable energy sources. When the renewable energy has surplus, the controller uses it to compensate the consumption of traditional sources. As long as the renewable energy supply and the power demand are balanced in the long run, i.e., the controller does not need to infuse more and more traditional energy into the grid in the long term, the controller is regarded as successful in operating a smart grid powered by renewable energy.

Therefore, the goal of the agent is to minimize the total loss $\sum_{t=0}^{T-1} f_t(\boldsymbol{x}_t)$ subject to the long-term constraint $\sum_{t=0}^{T-1} \boldsymbol{g}_t(\boldsymbol{x}_t) \preceq \boldsymbol{0}$, which can be casted into the following optimization problem:

$$\begin{aligned} &\text{Minimize}_{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{T-1} \in \mathbb{X}} \sum_{t=0}^{T-1} f_t(\boldsymbol{x}_t) \\ &\text{subject to} \sum_{t=0}^{T-1} \boldsymbol{g}_t(\boldsymbol{x}_t) \preceq \boldsymbol{0}. \end{aligned} \quad (2)$$

Solving problem (2) exactly is still impossible in the online setting here, because the information about the loss functions and constraint functions are unknown in advance. Instead, our goal is to obtain a total loss $\sum_{t=0}^{T-1} f_t(\boldsymbol{x}_t)$ that is not too large compared to some benchmark, and meanwhile, to ensure that $\sum_{t=0}^{T-1} \boldsymbol{g}_t(\boldsymbol{x}_t)$ is "not too positive", i.e., the long-term constraint is not violated too much. Here, different from the static benchmark used in [11], [19], [20], we choose $\{\boldsymbol{x}_t^*\}_{t=0}^{\infty}$ as a *dynamic* benchmark sequence. We note that dynamic benchmark is more meaningful than static benchmark when the underlying optima of the system is inherently changing, e.g., tracking a moving target. The first performance criterion is the regret of the objective function with respect to the dynamic benchmark as follows:

$$\texttt{Reg}(T) := \sum_{t=0}^{T-1} \left[f_t(\boldsymbol{x}_t) - f_t(\boldsymbol{x}_t^*)\right]. \quad (3)$$

The second performance metric is the constraint violations:

$$\texttt{Vio}_k(T) := \sum_{t=0}^{T-1} g_{t,k}(\boldsymbol{x}_t),\ k = 1, \ldots, m, \quad (4)$$

where $g_{t,k}(\cdot)$ is the k-th component of vector-valued constraint function $\boldsymbol{g}_t(\cdot)$, i.e., $\boldsymbol{g}_t(\boldsymbol{x}) = [g_{t,1}(\boldsymbol{x}), \ldots, g_{t,m}(\boldsymbol{x})]^{\mathsf{T}}$. Ideally, the regret and the constraint violations should be sublinear with respect to T, i.e.,[2] $\texttt{Reg}(T) \leq o(T)$ and $\texttt{Vio}_k(T) \leq o(T)$, $\forall k = 1, \ldots, m$. Hence, as T goes to infinity, $\frac{\texttt{Reg}(T)}{T} \leq o(1) \to 0$ and $\frac{\texttt{Vio}_k(T)}{T} \leq o(1) \to 0$. This means that, as the time length T increases to infinity, the time-average regret and the time-average constraint violations are non-positive so that the performance of the sequence $\{\boldsymbol{x}_t\}$ is no worse than that of the benchmark sequence $\{\boldsymbol{x}_t^*\}$ asymptotically.

B. Virtual-Queue-Based Algorithm

Before formally presenting the proposed algorithm, we first make several technical assumptions, which are useful in later analysis.

Assumption 1: For any $t \geq 0$ and $k = 1, \ldots, m$, f_t and $g_{t,k}$ are convex continuous functions.

Assumption 2: The action set $\mathbb{X}$ is closed and convex.

Assumption 3: There exists some $\beta > 0$ such that for every $t = 0, 1, \ldots$, $\boldsymbol{g}_t$ is Lipschitz continuous with modulous β, i.e., $\|\boldsymbol{g}_t(\boldsymbol{x}) - \boldsymbol{g}_t(\boldsymbol{y})\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_2$, for any $\boldsymbol{x}, \boldsymbol{y}$.

Assumption 4: There exists some $R > 0$ such that $\|\boldsymbol{x}\|_2 \leq R$ for any $\boldsymbol{x} \in \mathbb{X}$.

Assumption 5: There exists some $F > 0$ such that $|f_t(\boldsymbol{x})| \leq F$ for any $t \geq 0, \boldsymbol{x} \in \mathbb{X}$.

[2] We follow the standard definition of $o(\cdot)$. For positive sequences a_n and b_n, $a_n = o(b_n)$ means $\lim_{n\to\infty} \frac{a_n}{b_n} = 0$.

Algorithm 1: An Online Algorithm Based on Virtue Queues.

Inputs:

Cost function and constraint function sequences: $\{f_t, \boldsymbol{g}_t\}_{t=0}^{\infty}$

Outputs:

Action sequence: $\{\boldsymbol{x}_t\}_{t=0}^{\infty}$

1: Initialize $\boldsymbol{x}_0 \in \mathbb{X}$, $\boldsymbol{Q}_0 = \boldsymbol{0}$, $\boldsymbol{g}_{-1}(\cdot) \equiv \boldsymbol{0}$.

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

3: Update the action:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}\in\mathbb{X}} \Big\{ f_t(\boldsymbol{x}) + (\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t))^{\mathsf{T}} \boldsymbol{g}_t(\boldsymbol{x}) + \alpha \|\boldsymbol{x} - \boldsymbol{x}_t\|_2^2 \Big\}. \quad (5)$$

4: Update the virtual queues: for $k = 1, \ldots, m$,

$$Q_{t+1,k} = \max\{-g_{t,k}(\boldsymbol{x}_{t+1}), Q_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1})\}. \quad (6)$$

5: **end for**

Assumption 6: There exists some $G > 0$ such that $\|\boldsymbol{g}_t(\boldsymbol{x})\|_2 \leq G$ for any $t \geq 0, \boldsymbol{x} \in \mathbb{X}$.

We note that the aforementioned assumptions are all standard in the literature of OCO [2], [3]. Specifically, Assumptions 1, 2, 3 are standard convexity and Lipschitz continutiy assumptions widely used in the analysis of primal-dual algorithms such as saddle point methods [19], [24] and ADMM [5], [29], [30]. Assumptions 4, 5, 6 bound the impact of individual actions and function outputs so that one single term at one time instant is not disastrous for the regret and constraint violations.

Now, we propose an online optimization algorithm based on virtual queues as shown in Algorithm 1, in which $\alpha > 0$ is an algorithm parameter and satisfies the following assumption.

Assumption 7: The algorithm parameter α and Lipschitz modulus β satisfy $\alpha \geq \beta^2$.

Assumption 7 requires that the algorithm stepsize parameter α is large enough, i.e., the actions evolve slowly enough so that there is no abrupt change. We note that Algorithm 1 is an online version of the optimization algorithm in [1] when the objective functions and constraint functions are time-varying and unknown a priori. This extension to online setting necessitates judicious choice of the time indicies of functions in the algorithm. In particular, the $\boldsymbol{x}$-update in (5) makes use of both functions $\boldsymbol{g}_t$ and $\boldsymbol{g}_{t-1}$ instead of only $\boldsymbol{g}_t$. This subtle difference is important for later performance analysis.

Besides the action sequence $\{\boldsymbol{x}_t\}$, Algorithm 1 also maintains and updates a sequence of auxiliary variables $\{\boldsymbol{Q}_t\}$, which are called virtual queues. Actually, in (6), if the first term in the maximization is replaced by 0, the update of $\boldsymbol{Q}_t$ is tantamount to the standard queue updates with increments $\boldsymbol{g}_t(\boldsymbol{x}_{t+1})$. Thus, the virtual queues $\{\boldsymbol{Q}_t\}$ characterize the cumulative constraint violations and bounds of the queue backlogs can be readily translated to bounds of constraint violations. In adaptive signal processing, (5) corresponds to the updates of the estimates of the unknown parameters and (6) corresponds to the updates of the auxiliary variables. Together, as data arrive sequentially, these two iterative equations enable real-time tracking of the possibly time-varying unknown parameters.

Algorithm 1 has close connection with the saddle point type of methods widely used in the literature of constrained OCO [19], [20], [24], [26], if $\{\boldsymbol{x}_t\}$ is viewed as primal variables and $\{\boldsymbol{Q}_t\}$ is regarded as dual variables (Lagrange multipliers). The main difference between Algorithm 1 and the saddle point methods is on the updates of dual variables and the way of incorporating constraint functions in the (modified) Lagrangian. These differences render Algorithm 1 some advantages over saddle point methods in terms of performance guarantees, which will be elaborated in Section III in more detail. Further, we note that an online algorithm for constrained OCO has been proposed in [27] recently, which is also based on virtual queues. Nevertheless, the specific updates of the algorithm in [27] are very different from those in Algorithm 1. Besides, a static offline optimum is used to define the regret in [27], in contrast to the more practically meaningful dynamic benchmark sequence $\{\boldsymbol{x}_t^*\}$ adopted in this paper.

It has been shown in [1] that a static form of Algorithm 1 converges to the optimal point with a rate of $\mathcal{O}\left(\frac{1}{t}\right)$ when the objective and constraint functions are time-invariant. A main goal of the current paper is to examine the impact of time-varying objective and constraint functions, which are unknown until the corresponding actions have been chosen. Specifically, our goal is to ascertain (i) the impact of temporal variations of the dynamic optimization problems on the performance guarantees of the regret and constraint violations of Algorithm 1; (ii) under what conditions can we guarantee sublinear regret and constraint violations for Algorithm 1. Intuitively, the performance of Algorithm 1 should depend on how drastically $\{f_t\}$ and $\{\boldsymbol{g}_t\}$ vary across time. In Section III, we quantify this dependence by deriving explicit upper bounds on the regret and constraint violations in terms of temporal variations of $\{f_t\}$ and $\{\boldsymbol{g}_t\}$. Finally, we remark that Algorithm 1 can be implemented (and may work well) even if all of Assumptions 1-7 do not hold. These assumptions are made to facilitate performance analysis. We may not guarantee "good" performance of Algorithm 1 theoretically without these assumptions, which are all standard in the performance analysis of OCO algorithms in the literature.

III. Performance Analysis

In this section, under Assumptions 1-7, we analyze the performance of Algorithm 1 by deriving upper bounds on its regret and constraint violations. The impact of the temporal variation of the dynamic optimization problem (1) on the regret and constraint violations of Algorithm 1 is explicitly demonstrated. Based on the performance bounds, we obtain sufficient conditions under which Algorithm 1 possesses sublinear regret and sublinear constraint violations. Finally, the theoretical advantages of Algorithm 1 over MOSP in [26] are discussed. In the following, we carry out detailed performance analysis of Algorithm 1. We begin with some simple relations for the virtual queues.

Lemma 1: 1) For any $t \geq 0, k = 1, \ldots, m$: $Q_{t,k} \geq 0$.

2) For any $t \geq -1$, $k = 1, \ldots, m$: $Q_{t+1,k} + g_{t,k}(\boldsymbol{x}_{t+1}) \geq 0$.

3) For any $t \geq 0$: $\|\boldsymbol{Q}_{t+1}\|_2 \geq \|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2$.

Proof: The proof is presented in Appendix A. ■

For $t \geq 0$, as $\boldsymbol{Q}_t$ is a virtual queue, we can further define a Lyapunov function $L_t := \frac{1}{2}\|\boldsymbol{Q}_t\|_2^2$ to quantify the size of the queue backlog. In addition, we define the Lyapunov drift $\Delta_t := L_{t+1} - L_t$ to describe the evolution of the queue backlog. Then, we have the following bound for the Lyapunov drift.

Lemma 2: For every $t \geq 0$, we have:

$$\Delta_t \leq \boldsymbol{Q}_t^{\mathsf{T}} \boldsymbol{g}_t(\boldsymbol{x}_{t+1}) + \|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2. \tag{7}$$

Proof: The proof is presented in Appendix B. ■

To achieve a joint bound for the incurred cost and Lyapunov drift (which is related to the constraint violations due to the update of the queue backlogs in (6)) by Algorithm 1, we derive the following result.

Lemma 3: For $t \geq 0$, we have:

$$\Delta_t + f_t(\boldsymbol{x}_{t+1}) \leq f_t(\boldsymbol{x}_t^*) + \alpha\left(\|\boldsymbol{x}_t - \boldsymbol{x}_t^*\|_2^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2\right)$$
$$+\frac{1}{2}\left(\|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 - \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2\right) + \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t)\|_2^2. \tag{8}$$

Proof: The proof is presented in Appendix C. ■

The goal of the performance analysis is to study the impact of the temporal variations of the dynamic optimization problem (1) on the regret and constraint violations of Algorithm 1. To this end, we need to quantify the temporal variations of problem (1). In particular, we need to quantify the temporal variations of function sequences. To this end, for any positive integer l, n and compact (i.e., closed and bounded) set $\mathbb{X} \subset \mathbb{R}^n$, we define a linear space:

$$\mathcal{C}_l(\mathbb{X}) := \left\{\boldsymbol{\phi} : \mathbb{X} \mapsto \mathbb{R}^l \,\middle|\, \boldsymbol{\phi} \text{ is continuous on } \mathbb{X}\right\}, \tag{9}$$

which is equipped with a norm $\|\cdot\|_\infty$ defined as:

$$\|\boldsymbol{\phi}\|_\infty = \max_{\boldsymbol{x} \in \mathbb{X}} \|\boldsymbol{\phi}(\boldsymbol{x})\|_2, \ \ \forall \boldsymbol{\phi} \in \mathcal{C}_l(\mathbb{X}). \tag{10}$$

Note that when $l = 1$, (10) degenerates to:

$$\|\phi\|_\infty = \max_{\boldsymbol{x} \in \mathbb{X}} |\phi(\boldsymbol{x})|, \ \ \forall \phi \in \mathcal{C}_1(\mathbb{X}). \tag{11}$$

Now, we can define the total variations of the point sequence $\{\boldsymbol{x}_t^*\}$ and the function sequence $\{f_t\} \subset \mathcal{C}_1(\mathbb{X})$ as:

$$V_{\boldsymbol{x}}(t) := \sum_{\tau=0}^{t-1} \left\|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\right\|_2, \tag{12}$$

$$V_f(t) := \sum_{\tau=0}^{t-1} \|f_\tau - f_{\tau+1}\|_\infty. \tag{13}$$

Furthermore, we define the total squared variation of the function sequence $\{\boldsymbol{g}_t\} \subset \mathcal{C}_m(\mathbb{X})$ as:

$$V_{\boldsymbol{g}}(t) := \sum_{\tau=1}^{t-1} \|\boldsymbol{g}_{\tau-1} - \boldsymbol{g}_\tau\|_\infty^2. \tag{14}$$

Another related quantity that will be used in later analysis (Theorem 2 for constraint violations) is the total variation of $\{\boldsymbol{g}_t\}$ defined as:

$$\widetilde{V}_{\boldsymbol{g}}(t) := \sum_{\tau=1}^{t-1} \|\boldsymbol{g}_{\tau-1} - \boldsymbol{g}_\tau\|_\infty. \tag{15}$$

Both $V_{\boldsymbol{g}}(t)$ and $\widetilde{V}_{\boldsymbol{g}}(t)$ characterize the temporal evolution of $\{\boldsymbol{g}_t\}$. We observe that, in many practical cases, if $V_{\boldsymbol{g}}(t)$ or $\widetilde{V}_{\boldsymbol{g}}(t)$ is sublinear, then $V_{\boldsymbol{g}}(t)$ is often smaller than $\widetilde{V}_{\boldsymbol{g}}(t)$ in the order sense. The reason is as follows. If $\|\boldsymbol{g}_{t-1} - \boldsymbol{g}_t\|_\infty$ is a constant, then both $V_{\boldsymbol{g}}(t)$ and $\widetilde{V}_{\boldsymbol{g}}(t)$ are exactly linear in t. So, when $V_{\boldsymbol{g}}(t)$ or $\widetilde{V}_{\boldsymbol{g}}(t)$ is sublinear, it is often the case (except for some strange sequences constructed on purpose) that $\|\boldsymbol{g}_{t-1} - \boldsymbol{g}_t\|_\infty$ converges to zero. Hence, $\|\boldsymbol{g}_{t-1} - \boldsymbol{g}_t\|_\infty$ is much smaller than 1 for large t and $V_{\boldsymbol{g}}(t)$ is smaller than $\widetilde{V}_{\boldsymbol{g}}(t)$ in the order sense. For instance, if $\|\boldsymbol{g}_{t-1} - \boldsymbol{g}_t\|_\infty \propto t^a$ for some $a \in \mathbb{R}$, then $V_g(t) = \mathcal{O}\left(t^{2a+1}\right)$ and $\widetilde{V}_g(t) = \mathcal{O}\left(t^{a+1}\right)$. So the condition for a sublinear $V_{\boldsymbol{g}}(t)$ or $\widetilde{V}_{\boldsymbol{g}}(t)$ is $a < 0$. If so, $V_g(t)$ is smaller than $\widetilde{V}_{\boldsymbol{g}}(t)$ in the order sense.

Next, we have the following bound closely related to the regret and the queue backlogs (and thus related to the constraint violations) in terms of the total variations or total squared variations defined above.

Lemma 4: For any $t \geq 1$, we have:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau)$$
$$\leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) + \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 - \frac{1}{2}\|\boldsymbol{Q}_t\|_2^2 + V_f(t) + V_{\boldsymbol{g}}(t)$$
$$+ 4\alpha R V_{\boldsymbol{x}}(t) + \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + F + f_0(\boldsymbol{x}_0) + \alpha\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2. \tag{16}$$

Proof: Replace t by τ in (8). For arbitrary $t \geq 1$, summing (8) for τ from 0 to $t-1$, we obtain:

$$\sum_{\tau=0}^{t-1} \Delta_\tau + \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_{\tau+1})$$
$$\leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) + \alpha \sum_{\tau=0}^{t-1}\left(\|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^*\|_2^2 - \|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_\tau^*\|_2^2\right)$$
$$+ \frac{1}{2}\sum_{\tau=0}^{t-1}\left(\|\boldsymbol{g}_\tau(\boldsymbol{x}_{\tau+1})\|_2^2 - \|\boldsymbol{g}_{\tau-1}(\boldsymbol{x}_\tau)\|_2^2\right)$$
$$+ \sum_{\tau=0}^{t-1} \|\boldsymbol{g}_{\tau-1}(\boldsymbol{x}_\tau) - \boldsymbol{g}_\tau(\boldsymbol{x}_\tau)\|_2^2. \tag{17}$$

We note:

$$\|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^*\|_2^2 - \|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_\tau^*\|_2^2$$
$$= \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^*\|_2^2 - \|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_{\tau+1}^* + \boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2^2 \tag{18}$$

$$\overset{(a)}{\leq} \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^*\|_2^2 - \Big(\|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_{\tau+1}^*\|_2^2 - 2\|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_{\tau+1}^*\|_2 \|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2 + \|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2^2\Big) \quad (19)$$

$$\overset{(b)}{\leq} \|\boldsymbol{x}_\tau - \boldsymbol{x}_\tau^*\|_2^2 - \|\boldsymbol{x}_{\tau+1} - \boldsymbol{x}_{\tau+1}^*\|_2^2 + 4R\|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2, \quad (20)$$

where in (a) we make use of $\|\boldsymbol{u}+\boldsymbol{v}\|_2^2 \geq \|\boldsymbol{u}\|_2^2 - 2\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2 + \|\boldsymbol{v}\|_2^2, \forall \boldsymbol{u}, \boldsymbol{v}$; in (b), we use Assumption 4. Thus, from (17), by making use of (20) and the definition of Lyapunov drift, we obtain:

$$L_t - L_0 + \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_{\tau+1}) \leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) + \alpha\Bigg(\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{x}_t^*\|_2^2 + 4R\sum_{\tau=0}^{t-1}\|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2\Bigg) + \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 + \sum_{\tau=0}^{t-1}\|\boldsymbol{g}_{\tau-1}(\boldsymbol{x}_\tau) - \boldsymbol{g}_\tau(\boldsymbol{x}_\tau)\|_2^2. \quad (21)$$

We further note:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_{\tau+1}) = \sum_{\tau=0}^{t-1}[f_{\tau+1}(\boldsymbol{x}_{\tau+1}) + f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})] \quad (22)$$

$$\geq \sum_{\tau=1}^{t} f_\tau(\boldsymbol{x}_\tau) - \sum_{\tau=0}^{t-1}|f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})| \quad (23)$$

$$\overset{(a)}{\geq} \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau) - F - f_0(\boldsymbol{x}_0) - \sum_{\tau=0}^{t-1}|f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})|, \quad (24)$$

where in (a) we make use of Assumption 5. Substituting (24) into (21) and noting that $L_t = \frac{1}{2}\|\boldsymbol{Q}_t\|_2^2$ and $L_0 = 0$, we get:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau) \leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) + \alpha(\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{x}_t^*\|_2^2) + \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 - \frac{1}{2}\|\boldsymbol{Q}_t\|_2^2 + \sum_{\tau=0}^{t-1}|f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})| + \sum_{\tau=0}^{t-1}\|\boldsymbol{g}_{\tau-1}(\boldsymbol{x}_\tau) - \boldsymbol{g}_\tau(\boldsymbol{x}_\tau)\|_2^2 + 4\alpha R\sum_{\tau=0}^{t-1}\|\boldsymbol{x}_{\tau+1}^* - \boldsymbol{x}_\tau^*\|_2 + F + f_0(\boldsymbol{x}_0). \quad (25)$$

In addition, we have:

$$\sum_{\tau=0}^{t-1}|f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})| \leq \sum_{\tau=0}^{t-1}\|f_\tau - f_{\tau+1}\|_\infty = V_f(t), \quad (26)$$

and

$$\sum_{\tau=0}^{t-1}\|\boldsymbol{g}_{\tau-1}(\boldsymbol{x}_\tau) - \boldsymbol{g}_\tau(\boldsymbol{x}_\tau)\|_2^2 \leq \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + \sum_{\tau=1}^{t-1}\|\boldsymbol{g}_{\tau-1} - \boldsymbol{g}_\tau\|_\infty^2 = \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + V_{\boldsymbol{g}}(t). \quad (27)$$

Substituting (26), (27) into (25), making use of the definition of $V_{\boldsymbol{x}}$ in (12), and noting that $\|\boldsymbol{x}_t - \boldsymbol{x}_t^*\|_2^2 \geq 0$, we achieve the result in (16). ∎

Next, we are ready to present the first main result regarding the regret of Algorithm 1.

Theorem 1: (Regret) For any $t \geq 1$, we have:

$$\mathtt{Reg}(t) \leq V_f(t) + V_{\boldsymbol{g}}(t) + 4\alpha R V_{\boldsymbol{x}}(t) + \alpha\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2 + \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + F + f_0(\boldsymbol{x}_0). \quad (28)$$

In particular, if $V_f, V_{\boldsymbol{g}}, V_{\boldsymbol{x}}$ are sublinear in t, then $\mathtt{Reg}(t)$ is also sublinear, i.e., $\mathtt{Reg}(t) \leq o(t)$.

Proof: By Lemma 1-(3), we know that $\|\boldsymbol{Q}_t\|_2 \geq \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2, \forall t \geq 1$. Substituting this inequality into (16) gives (28). ∎

Next, we endeavor to bound the constraint violations $\mathtt{Vio}_k$ of Algorithm 1. To this end, we first link the constraint violations with the queue backlogs. By the queue update in (6), for any $t \geq 1$, we have:

$$Q_{\tau+1,k} \geq Q_{\tau,k} + g_{\tau,k}(\boldsymbol{x}_{\tau+1}), \quad \forall \tau = 0, \ldots, t-1. \quad (29)$$

Summing (29) over all $\tau = 0, \ldots, t-1$ leads to the following lemma.

Lemma 5: For any $t \geq 1$ and any $k = 1, \ldots, m$, we have:

$$Q_{t,k} \geq \sum_{\tau=0}^{t-1} g_{\tau,k}(\boldsymbol{x}_{\tau+1}). \quad (30)$$

Denote the dual optimal point of the optimization problem (1) as $\boldsymbol{\lambda}_t^* \in \mathbb{R}^m$. We further define the total variation of the dual optimal points $\{\boldsymbol{\lambda}_t^*\}$ as:

$$V_{\boldsymbol{\lambda}}(t) := \sum_{\tau=0}^{t-2}\left\|\boldsymbol{\lambda}_\tau^* - \boldsymbol{\lambda}_{\tau+1}^*\right\|_2, \ \ \forall t \geq 1. \quad (31)$$

According to Lemma 5, to bound the constraint violations, we only need to bound the queue backlog $\boldsymbol{Q}_t$, which is accomplished in the following lemma.

Lemma 6: For any $t \geq 1$, we have:

$$\|\boldsymbol{Q}_t\|_2 \leq 4V_{\boldsymbol{\lambda}}(t) + 2\sqrt{V_f(t)} + \sqrt{2V_{\boldsymbol{g}}(t)} + \sqrt{8\alpha R V_{\boldsymbol{x}}(t)} + \sqrt{G^2 + 4F + 2\alpha\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2 + 2\|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2} + 2\left\|\boldsymbol{\lambda}_0^*\right\|_2. \quad (32)$$

Proof: For $t \geq 0$, the dual function of problem (1) is:

$$q_t(\boldsymbol{\lambda}) = \inf_{\boldsymbol{x}\in\mathbb{X}} \left\{ f_t(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathsf{T}} \boldsymbol{g}_t(\boldsymbol{x}) \right\}. \tag{33}$$

According to the strong duality of the convex optimization problem (1), we have:

$$f_t(\boldsymbol{x}_t^*) = q_t(\boldsymbol{\lambda}_t^*) \tag{34}$$

$$\leq f_t(\boldsymbol{x}_{t+1}) + \boldsymbol{\lambda}_t^{*\mathsf{T}} \boldsymbol{g}_t(\boldsymbol{x}_{t+1}) \tag{35}$$

$$\overset{\text{(a)}}{\leq} f_t(\boldsymbol{x}_{t+1}) + \boldsymbol{\lambda}_t^{*\mathsf{T}} (\boldsymbol{Q}_{t+1} - \boldsymbol{Q}_t), \tag{36}$$

where in (a) we make use of $\boldsymbol{g}_t(\boldsymbol{x}_{t+1}) \preceq \boldsymbol{Q}_{t+1} - \boldsymbol{Q}_t$ (c.f. (6)) and $\boldsymbol{\lambda}_t^* \succeq \mathbf{0}$. Substitute t by τ in (36). For arbitrary $t \geq 1$, summing (36) for τ from 0 to $t-1$, we obtain:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) \leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_{\tau+1}) + \sum_{\tau=0}^{t-1} \boldsymbol{\lambda}_\tau^{*\mathsf{T}} (\boldsymbol{Q}_{\tau+1} - \boldsymbol{Q}_\tau). \tag{37}$$

In addition, from the definition of $V_{\boldsymbol{\lambda}}$ in (31), we have $V_{\boldsymbol{\lambda}}(t) \geq \|\boldsymbol{\lambda}_{t-1}^* - \boldsymbol{\lambda}_0^*\|_2 \geq \|\boldsymbol{\lambda}_{t-1}^*\|_2 - \|\boldsymbol{\lambda}_0^*\|_2$. Thus, we know:

$$\|\boldsymbol{\lambda}_{t-1}^*\|_2 \leq V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2. \tag{38}$$

Hence,

$$\sum_{\tau=0}^{t-1} \boldsymbol{\lambda}_\tau^{*\mathsf{T}} (\boldsymbol{Q}_{\tau+1} - \boldsymbol{Q}_\tau) = \boldsymbol{\lambda}_{t-1}^{*\mathsf{T}} \boldsymbol{Q}_t + \sum_{\tau=0}^{t-2} \left(\boldsymbol{\lambda}_\tau^* - \boldsymbol{\lambda}_{\tau+1}^*\right)^{\mathsf{T}} \boldsymbol{Q}_{\tau+1} \tag{39}$$

$$\leq \left\|\boldsymbol{\lambda}_{t-1}^*\right\|_2 \|\boldsymbol{Q}_t\|_2 + \sum_{\tau=0}^{t-2} \left\|\boldsymbol{\lambda}_\tau^* - \boldsymbol{\lambda}_{\tau+1}^*\right\|_2 \|\boldsymbol{Q}_{\tau+1}\|_2 \tag{40}$$

$$\overset{\text{(a)}}{\leq} (V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2)\|\boldsymbol{Q}_t\|_2 + V_{\boldsymbol{\lambda}}(t) \max_{\tau=1,\ldots,t} \|\boldsymbol{Q}_\tau\|_2 \tag{41}$$

$$\leq (2V_{\boldsymbol{\lambda}} + \|\boldsymbol{\lambda}_0^*\|_2) \max_{\tau=1,\ldots,t} \|\boldsymbol{Q}_\tau\|_2, \tag{42}$$

where in (a) we make use of (38). Moreover, we have:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_{\tau+1})$$

$$= \sum_{\tau=0}^{t-1} [f_{\tau+1}(\boldsymbol{x}_{\tau+1}) + f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})] \tag{43}$$

$$\leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau) + f_t(\boldsymbol{x}_t) - f_0(\boldsymbol{x}_0) + \sum_{\tau=0}^{t-1} |f_\tau(\boldsymbol{x}_{\tau+1}) - f_{\tau+1}(\boldsymbol{x}_{\tau+1})| \tag{44}$$

$$\leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau) + F - f_0(\boldsymbol{x}_0) + V_f(t). \tag{45}$$

Substituting (42) and (45) into (37), we get, for any $t \geq 1$:

$$\sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau^*) \leq \sum_{\tau=0}^{t-1} f_\tau(\boldsymbol{x}_\tau) + F - f_0(\boldsymbol{x}_0) + V_f(t) + (2V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2) \max_{\tau=1,\ldots,t} \|\boldsymbol{Q}_\tau\|_2. \tag{46}$$

Define $l(t) = \arg\max_{\tau=1,\ldots,t} \|\boldsymbol{Q}_\tau\|_2$. Substituting t by $l(t)$ in (16) and making use of Assumption 6, we achieve:

$$\begin{aligned}\sum_{\tau=0}^{l(t)-1} f_\tau(\boldsymbol{x}_\tau) &\leq \sum_{\tau=0}^{l(t)-1} f_\tau(\boldsymbol{x}_\tau^*) + \frac{1}{2}G^2 - \frac{1}{2}\left\|\boldsymbol{Q}_{l(t)}\right\|_2^2 + V_f(l(t)) + V_{\boldsymbol{g}}(l(t)) \\ &\quad + 4\alpha R V_{\boldsymbol{x}}(l(t)) + \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + F \\ &\quad + f_0(\boldsymbol{x}_0) + \alpha\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2.\end{aligned} \tag{47}$$

Substituting t by $l(t)$ in (46) and noting that $\max_{\tau=1,\ldots,l(t)} \|\boldsymbol{Q}_\tau\|_2 = \left\|\boldsymbol{Q}_{l(t)}\right\|_2$, we obtain:

$$\sum_{\tau=0}^{l(t)-1} f_\tau(\boldsymbol{x}_\tau^*) \leq \sum_{\tau=0}^{l(t)-1} f_\tau(\boldsymbol{x}_\tau) + F - f_0(\boldsymbol{x}_0) + V_f(l(t)) + (2V_{\boldsymbol{\lambda}}(l(t)) + \|\boldsymbol{\lambda}_0^*\|_2) \left\|\boldsymbol{Q}_{l(t)}\right\|_2. \tag{48}$$

Adding (47) and (48) together yields:

$$\begin{aligned}0 \leq &-\frac{1}{2}\left\|\boldsymbol{Q}_{l(t)}\right\|_2^2 + (2V_{\boldsymbol{\lambda}}(l(t)) + \|\boldsymbol{\lambda}_0^*\|_2)\left\|\boldsymbol{Q}_{l(t)}\right\|_2 \\ &+ 2V_f(l(t)) + V_{\boldsymbol{g}}(l(t)) + 4\alpha R V_{\boldsymbol{x}}(l(t)) \\ &+ \alpha\left\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\right\|_2^2 + \frac{1}{2}G^2 + \|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2 + 2F.\end{aligned} \tag{49}$$

Rearranging terms and noting that $V_{\boldsymbol{x}}, V_f, V_{\boldsymbol{g}}, V_{\boldsymbol{\lambda}}$ are all monotonically increasing sequences, we have:

$$\begin{aligned}&\left[\left\|\boldsymbol{Q}_{l(t)}\right\|_2 - (2V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2)\right]^2 \\ &\leq (2V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|)^2 + 4V_f(t) + 2V_{\boldsymbol{g}}(t) + 8\alpha R V_{\boldsymbol{x}}(t) \\ &\quad + G^2 + 4F + 2\alpha\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\|_2^2 + 2\|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2.\end{aligned} \tag{50}$$

Hence,

$$\|\boldsymbol{Q}_t\|_2 \leq \left\|\boldsymbol{Q}_{l(t)}\right\|_2 \tag{51}$$

$$\leq \left|\left\|\boldsymbol{Q}_{l(t)}\right\|_2 - (2V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2)\right| + 2V_{\boldsymbol{\lambda}}(t) + \|\boldsymbol{\lambda}_0^*\|_2 \tag{52}$$

$$\begin{aligned}&\overset{\text{(a)}}{\leq} 4V_{\boldsymbol{\lambda}}(t) + 2\left\|\boldsymbol{\lambda}_0^*\right\|_2 + 2\sqrt{V_f(t)} + \sqrt{2V_{\boldsymbol{g}}(t)} + \sqrt{8\alpha R V_{\boldsymbol{x}}(t)} \\ &\quad + \sqrt{G^2 + 4F + 2\alpha\left\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\right\|_2^2 + 2\|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2},\end{aligned} \tag{53}$$

where in (a) we exploit (50) and make use of the fact $\sqrt{\sum_{i=1}^{I} a_i} \leq \sum_{i=1}^{I} \sqrt{a_i},\ \forall a_i \geq 0, i = 1, \ldots, I.$ ∎

Next, we are ready to present our second main result regarding the constraint violations of Algorithm 1.

Theorem 2: (Constraint Violations) For any $t \geq 1$ and $k = 1, \ldots, m$:

$$\begin{aligned}
&\mathtt{Vio}_k(t) \\
&\leq \widetilde{V}_{\boldsymbol{g}}(t) + 4V_{\boldsymbol{\lambda}}(t) + 2\sqrt{V_f(t)} + \sqrt{2V_{\boldsymbol{g}}(t)} + \sqrt{8\alpha R V_{\boldsymbol{x}}(t)} \\
&\quad + \sqrt{G^2 + 4F + 2\alpha \left\|\boldsymbol{x}_0 - \boldsymbol{x}_0^*\right\|_2^2 + 2\|\boldsymbol{g}_0(\boldsymbol{x}_0)\|_2^2} \\
&\quad + 2\left\|\boldsymbol{\lambda}_0^*\right\|_2 + g_{0,k}(\boldsymbol{x}_0). \qquad (54)
\end{aligned}$$

In particular, if $\widetilde{V}_{\boldsymbol{g}}$, $V_{\boldsymbol{\lambda}}$ are sublinear ($o(t)$) and V_f, $V_{\boldsymbol{g}}$, $V_{\boldsymbol{x}}$ are subquadratic ($o\left(t^2\right)$), then $\mathtt{Vio}_k(t)$ is sublinear, i.e., $\mathtt{Vio}_k(t) \leq o(t)$, for any $k = 1, \ldots, m$.

Proof: From Lemma 5, we know that for any $t \geq 2$, $k = 1, \ldots, m$:

$$Q_{t-1,k} \geq \sum_{\tau=0}^{t-2} g_{\tau,k}(\boldsymbol{x}_{\tau+1}). \qquad (55)$$

Therefore,

$$\begin{aligned}
\sum_{\tau=0}^{t-1} g_{\tau,k}(\boldsymbol{x}_\tau) &\leq g_{0,k}(\boldsymbol{x}_0) + \sum_{\tau=0}^{t-2} g_{\tau,k}(\boldsymbol{x}_{\tau+1}) \\
&\quad + \sum_{\tau=0}^{t-2} \left|g_{\tau+1,k}(\boldsymbol{x}_{\tau+1}) - g_{\tau,k}(\boldsymbol{x}_{\tau+1})\right| \qquad (56) \\
&\overset{\text{(a)}}{\leq} g_{0,k}(\boldsymbol{x}_0) + Q_{t-1,k} + \widetilde{V}_{\boldsymbol{g}}(t), \qquad (57)
\end{aligned}$$

where in (a) we make use of (55) and the definition of $\widetilde{V}_{\boldsymbol{g}}$ in (15). Note that the inequality (57) holds trivially for $t = 1$ and thus it actually hods for any $t \geq 1$. Note that $Q_{t-1,k} \leq \|\boldsymbol{Q}_{t-1}\|_2$. Thus, for any $t \geq 1$ and any $k = 1, \ldots, m$:

$$\sum_{\tau=0}^{t-1} g_{\tau,k}(\boldsymbol{x}_\tau) \leq g_{0,k}(\boldsymbol{x}_0) + \widetilde{V}_{\boldsymbol{g}}(t) + \|\boldsymbol{Q}_{t-1}\|_2. \qquad (58)$$

Making use of (32) in Lemma 6 and noting that the R.H.S. of (32) is monotonically increasing with t, we obtain the bound in (54). ■

From Theorems 1 and 2, we know that sufficient conditions for sublinear regret and constraint violations are that the variations of problem data, e.g., $V_{\boldsymbol{x}}(t)$, are sublinear or subquadratic. These conditions can be justified as follows. In most applications of online optimization, the underlying system varies slowly across time. For instance, in adaptive signal processing [28], it is commonly assumed that the unknown parameters (weight vectors) vary slowly across time. Otherwise, if the unknown parameters vary too fast, virtually no adaptive algorithm can track them well due to lack of information. In short, the underlying system varies slowly in most applications of online optimization, and so do the loss/constraint functions. Thus, the variations of the problem data, e.g., $V_{\boldsymbol{x}}$, are often sublinear, i.e., $o(t)$. On the contrary, if the variations are not sublinear, it is hard to guarantee sublinear regret and constraint violations theoretically. The reason is as follows. If the (cumulative) variations are[3] $\Omega(t)$, i.e., at least of the order t, then the gaps between loss/constraint functions at adjacent time slots are at least $\Omega(1)$, i.e., the loss/constraint functions evolve at a constant rate at minimum. Recall that, in OCO, the loss function f_t and constraint function $\boldsymbol{g}_t$ are revealed after the action $\boldsymbol{x}_t$ is determined. Thus, after $\boldsymbol{x}_t$ is determined, in principle, we should be able to choose a new loss function f_t and a new constraint function $\boldsymbol{g}_t$ such that the optimal point $\boldsymbol{x}_t^*$ is at least $\Omega(1)$ distance away from the chosen action $\boldsymbol{x}_t$. This may occur if the loss/constraint functions are chosen by a non-oblivious adversary who observes the selected action $\boldsymbol{x}_t$. In such a case, $\boldsymbol{x}_t$ cannot track the benchmark $\boldsymbol{x}_t^*$ well and the regret/constraint violation are not guaranteed to be sublinear.

Furthermore, two remarks regarding the comparison between Algorithm 1 and existing algorithms for constrained OCO are presented in the following.

Remark 1: We compare the performance guarantees of Algorithm 1 in Theorems 1 and 2 with those of the MOSP method presented in [26]. By choosing appropriate stepsize parameters, the MOSP can benefit from knowing the temporal variations of the underlying dynamic optimization problems (e.g., $V_{\boldsymbol{x}}(t)$). Thus, the MOSP distinguishes two cases: with or without the knowledge of the variations of the optimization problems. In Algorithm 1, the choice of stepsize parameter α does not depend on the variations of problem data. Thus, a fair comparison benchmark of Algorithm 1 should be MOSP without knowledge of variations (henceforth simply MOSP). For MOSP, according to [26], the regret is upper bounded by $\mathcal{O}\left(t^{\frac{1}{3}} \max\left\{V_{\boldsymbol{x}}(t), \widetilde{V}_{\boldsymbol{g}}(t), t^{\frac{1}{3}}\right\}\right)$ (with appropriate adaption of notations) while the constraint violation is upper bounded by $\mathcal{O}\left(t^{\frac{2}{3}}\right)$. In general, one cannot compare the regret and constraint violation bounds for Algorithm 1 with those of MOSP since the comparison results rely on the specific values of the related variations, i.e., for some values of the variations, the performance bounds of Algorithm 1 are better, and for some other values of the variations, the performance bounds of MOSP can be better. Nevertheless, several prominent advantages of the performance bounds for Algorithm 1 can be highlighted as follows. First, the regret bound for MOSP is always no less than $\mathcal{O}\left(t^{\frac{2}{3}}\right)$ even when $f_t, \boldsymbol{g}_t$ are time-invariant, i.e., all the variations are zero. In contrast, according to Theorem 1, the regret bound for Algorithm 1 can decrease to $\mathcal{O}(1)$ smoothly as the temporal variations of $f_t, \boldsymbol{g}_t$ decrease. Second, in order to guarantee sublinear regret for MOSP, the variations need to satisfy $V_{\boldsymbol{x}}(t) = o\left(t^{\frac{2}{3}}\right)$ and $\widetilde{V}_{\boldsymbol{g}}(t) = o\left(t^{\frac{2}{3}}\right)$, while the condition for sublinear regret of Algorithm 1 is $V_f(t) = o(t)$, $V_{\boldsymbol{g}}(t) = o(t)$ and $V_{\boldsymbol{x}}(t) = o(t)$. The latter condition is usually easier to be satisfied than the former, i.e., Algorithm 1 possesses sublinear regret for a broader class of dynamic optimization problems than MOSP does. Third, the bound for constraint violation of MOSP is always $\mathcal{O}\left(t^{\frac{2}{3}}\right)$ regardless of the variations. In contrast, the constraint violation bound for Algorithm 1 can

[3] We follow the standard definition of $\Omega(\cdot)$. That is, for sequences a_n and b_n, $a_n = \Omega(b_n)$ means $b_n = \mathcal{O}(a_n)$.

decrease smoothly to $\mathcal{O}(1)$ if the variations are sufficiently small. In particular, when f_t and $\boldsymbol{g}_t$ are time-invariant, both regret and constraint violation of Algorithm 1 become $\mathcal{O}(1)$, which is equivalent to the $\mathcal{O}\left(\frac{1}{t}\right)$ convergence rate established in [1] for static optimization problems. In this sense, Theorems 1 and 2 encompass the results in [1] as special cases. To achieve a more quantitative comparison between the performance guarantees of Algorithm 1 and MOSP, we consider the case that all variation terms are less than $\mathcal{O}\left(t^{\delta}\right)$ for some $\delta \in [0,1]$. In such a case, the regret and constraint violation bounds for Algorithm 1 are both $\mathcal{O}\left(t^{\delta}\right)$ while the regret and constraint violation bounds for MOSP are $\mathcal{O}\left(t^{\max\left\{\delta,\frac{1}{3}\right\}+\frac{1}{3}}\right)$ and $\mathcal{O}\left(t^{\frac{2}{3}}\right)$, respectively. This indicates that Algorithm 1 always possesses better regret bound than MOSP does and the constraint violation bound of Algorithm 1 is better than that of MOSP when $\delta < \frac{2}{3}$, i.e., when the variations of the dynamic optimization problem are not too drastic. Finally, we remark that, different from the various saddle point methods widely used for constrained OCO [19], [23], [24], [26], in Theorems 1 and 2, the stepsize parameter α of Algorithm 1 does not rely on the total time horizon t, which may be unknown in practice, i.e., the online optimization procedure may terminate at some unknown time. This observation also implies that the performance bounds of Algorithm 1 in Theorems 1 and 2 hold for arbitrary time slots, inlcuding those before the termination of the online optimization procedure. In contrast, the performance bounds of saddle point methods generally only hold for the time slot when the online procedure is ended [19], [23], [24], [26].

Remark 2: Another related online algorithm for constrained OCO has been developed by Neely and Yu in [27] recently. Though the algorithm in [27] is also based on virtual queues, its specific updates are very different from those of Algorithm 1. Additionally, the performance criterion used in [27] is also different from that of this paper. Specifically, the regret defined in [27] is with respect to the *static* offline optima, while the regret used in this paper is with respect to the *dynamic* optimal benchmark sequence $\{\boldsymbol{x}_t^*\}$ (c.f. (3)). This renders the performance criterion of this paper more practically meaningful (yet more challenging to analyze) since a static optimum may not be a good benchmark if the underlying system is inherently time-varying. In [27], dynamic benchmark is used only for the special case of i.i.d. loss/constraint function sequences $\{f_t(\cdot), \boldsymbol{g}_t(\cdot)\}$. This i.i.d. assumption is often unrealistic because loss/constraint functions at different time slots can be highly correlated in most practical applications, e.g., these functions may evolve in a Markovian way in many scenarios. In light of the above, the performance analysis and guarantees in [27] are fundamentally different from those in this paper.

IV. Data Center Resource Allocation

In this section, we study the dynamic resource allocation problem in data centers [8]–[10], [26], [31] under a constrained OCO framework by invoking the proposed Algorithm 1. In such a case, Algorithm 1 only involves simple closed-form computation and is amenable to distributed parallel implementation. For comparison purposes, application of MOSP in [26] is also considered. Numerical experiments are conducted to corroborate the effectiveness of Algorithm 1 and its adavantage over MOSP.

A. Problem Formulation and Algorithm Development

Consider a cloud computing network comprised of J mapping nodes and K data centers. Each mapping node collects the data processing requests from a local region and then transmits them to the K data centers to accomplish the data processing tasks. At each time t, we denote the amount of data requests arriving at mapping node j as $d_{t,j}$. Then, each mapping node j sends $y_{t,jk}$ amount of data processing tasks to data center k at time t. Finally, each data center k accomplishes $z_{t,k}$ amount of data tasks at time t. Each mapping node and each data center have a local queue to buffer the unserved data requests. Denote the queue backlog at mapping node j at time t as $w_{t,j}$, $j = 1, \ldots, J$. Denote the queue backlog at data center k at time t as $w_{t,J+k}$, $k = 1, \ldots, K$. Define the queue backlog vector at time t as $\boldsymbol{w}_t = [w_{t,1}, \ldots, w_{t,J+K}]^{\mathsf{T}}$. Define the extended data arrival vector at time t as $\boldsymbol{d}_t = [d_{t,1}, \ldots, d_{t,J}, \boldsymbol{0}_{1\times K}]^{\mathsf{T}}$. The control variables (or actions in OCO's term) of this cloud computing system at time t are collected in the vector $\boldsymbol{x}_t = [y_{t,11}, \ldots, y_{t,JK}, z_{t,1}, \ldots, z_{t,K}]^{\mathsf{T}} \in \mathbb{R}^{JK+K}$. The control variables need to satisfy $y_{t,jk} \le \bar{y}_{jk}$ and $z_{t,k} \le \bar{z}_k$ for any j, k, t, where $\bar{y}_{jk}$ and $\bar{z}_k$ are the maximum transmission rate of link (j,k) and maximum data processing rate of data center k, respectively. Define $\bar{\boldsymbol{x}} = [\bar{y}_{11}, \ldots, \bar{y}_{JK}, \bar{z}_1, \ldots, \bar{z}_K]^{\mathsf{T}}$ and the action set $\mathbb{X} = \{\boldsymbol{x} | \boldsymbol{0} \preceq \boldsymbol{x} \preceq \bar{\boldsymbol{x}}\}$. Furthermore, we define a constant matrix $\boldsymbol{C}$ as:

$$\boldsymbol{C} = \left[\begin{array}{ccc:c} -\boldsymbol{1}_K^{\mathsf{T}} & & & \\ & \ddots & & \boldsymbol{0}_{J\times K} \\ & & -\boldsymbol{1}_K^{\mathsf{T}} & \\ \hdashline \boldsymbol{I}_K & \cdots & \boldsymbol{I}_K & -\boldsymbol{I}_K \end{array}\right] \in \mathbb{R}^{(J+K)\times(JK+K)}, \quad (59)$$

where $\boldsymbol{1}_K$ is a K dimensional column vector of all ones and $\boldsymbol{I}_K$ is the $K \times K$ identity matrix. Then, the queueing updates for $\boldsymbol{w}_t$ can be written compactly as:

$$\boldsymbol{w}_{t+1} = [\boldsymbol{w}_t + \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{d}_t]^+, \quad (60)$$

where $x^+ = \max\{x, 0\}$ is defined entrywise. Suppose the initial queue backlog is some given $\boldsymbol{w}_0 \succeq \boldsymbol{0}$ and our goal is to clear all backlogs at time T, i.e., $\boldsymbol{w}_T = \boldsymbol{0}$, while minimizing some cost function related to the transmission rates and processing rates $\{\boldsymbol{x}_t\}$. Here, we consider the following time-varying quadratic cost function at time t:

$$f_t(\boldsymbol{x}) = \sum_{j=1}^{J}\sum_{k=1}^{K} \xi_{t,jk} y_{jk}^2 + \sum_{k=1}^{K} \eta_{t,k} z_k^2, \quad (61)$$

where $\boldsymbol{x}$ collects all $\{y_{jk}\}$ and $\{z_k\}$; $\xi_{t,jk} > 0$ and $\eta_{t,k} > 0$ are time-varying cost parameters of data transmission and data processing. The costs of data transmission are time-varying since the quality of wireless links varies. The costs of data processing at data centers change across time because the energy prices and availability are time-varying, e.g., due to the penetration of intermittent renewable energy sources. Hence, the dynamic resouce allocation problem of the cloud network can be posed as the following OCO problem:

$$\begin{aligned}&\text{Minimize}_{\{\boldsymbol{x}_t\}_{t=0}^{T-1}\subset\mathbb{X},\{\boldsymbol{w}_t\}_{t=1}^{T}} \sum_{t=0}^{T-1} f_t(\boldsymbol{x}_t)\\ &\text{subject to}\, \boldsymbol{w}_{t+1}=[\boldsymbol{w}_t+\boldsymbol{C}\boldsymbol{x}_t+\boldsymbol{d}_t]^+,\, t=0,\ldots,T-1,\\ &\qquad \boldsymbol{w}_T=\boldsymbol{0}.\end{aligned}\tag{62}$$

The constraints in problem (62) are difficult to deal with as they are coupled across time. We note that a necessary condition for these constraints to hold is $\sum_{t=0}^{T-1}(\boldsymbol{C}\boldsymbol{x}_t+\boldsymbol{d}_t)\preceq\boldsymbol{0}$ [26], [31]. This motivates us to consider the following OCO problem instead:

$$\begin{aligned}&\text{Minimize}_{\{\boldsymbol{x}_t\}_{t=0}^{T-1}\subset\mathbb{X}} \sum_{t=0}^{T-1} f_t(\boldsymbol{x}_t)\\ &\text{subject to} \sum_{t=0}^{T-1} g_t(\boldsymbol{x}_t)\preceq\boldsymbol{0},\end{aligned}\tag{63}$$

where $g_t(\boldsymbol{x})=\boldsymbol{C}\boldsymbol{x}+\boldsymbol{d}_t$. Problem (63) is evidently in the standard form of problem (2) and thus can be solved by the proposed Algorithm 1. We remark that the problem formulation here is related to existing settings for data center resource allocation [26], [31], [32]. Specifically, data transmission costs (bandwidth prices) and data processing costs (energy prices) have also been considered in [32] to perform joint request mapping and response routing with distributed data centers. The resource allocation model in [32] is static (one-shot) and all the requests from the clients must be fulfilled in the scheduling problem. In contrast, the data center resource allocation in this paper is dynamic and temporarily unfulfilled tasks can be stored in the queues at the mapping nodes and data centers. This renders the dynamic problem formulation in this paper more flexible in scheduling deferrable (delay-tolerant) tasks in data centers. Additionally, we note that the setting for data center resource allocation in [31] is also dynamic. Nevertheless, the problem in [31] falls into the category of stochastic optimization, i.e., the objective and constraint functions are expectations of random variables, which is different from the deterministic setting used in this paper. The problem formulation of data center resource allocation in this paper follows closely from that of [26], though the online optimization algorithms used to solve the problem are very different. Moreover, we note that online approaches for scheduling data centers (such as Algorithm 1 advocated in this paper) can handle time-varying model parameters that are available *sequentially* during the process of the online algorithms, e.g., the dynamic requests $d_{t,j}$ and the dynamic cost coefficients $\xi_{t,jk}, \eta_{t,k}$ may not be available until time t due to the intermittence/uncertainty of user requests and renewable generation. On the contrary, offline approaches can only allocate resources in a *batch* mode, i.e., all the model parameters need to be available *a priori* before the start of the offline algorithms, which is not the case in many practical scenarios. This also indicates that online approaches are superior to offline approaches in terms of computational complexity. The former processes the sequential data (model parameters) slot-by-slot while the latter processes all the data at all time slots in a batch manner, which is much more computationally demanding or even prohibitive.

Due to the decoupled structure of the cost function f_t and the network structure (reflected in $\boldsymbol{C}$) of the constraint function $\boldsymbol{g}_t$, Algorithm 1 can be implemented in a distributed manner, which will be detailed in the following. Define a diagonal matrix $\boldsymbol{\Lambda}_t=\operatorname{diag}\{\xi_{t,11},\ldots,\xi_{t,JK},\eta_{t,1},\ldots,\eta_{t,K}\}\in\mathbb{R}^{(JK+K)\times(JK+K)}$. Thus, $f_t(\boldsymbol{x})=\boldsymbol{x}^\mathsf{T}\boldsymbol{\Lambda}_t\boldsymbol{x}$. Denote $\boldsymbol{Q}_t\in\mathbb{R}^{J+K}$ as the virtual queues. Based on (5), the update of the actions can be written as:

$$\begin{aligned}\boldsymbol{x}_{t+1}=\arg\min_{\boldsymbol{0}\preceq\boldsymbol{x}\preceq\bar{\boldsymbol{x}}}\Big\{&\boldsymbol{x}^\mathsf{T}(\boldsymbol{\Lambda}_t+\alpha\boldsymbol{I})\boldsymbol{x}\\ &+\boldsymbol{x}^\mathsf{T}\left[\boldsymbol{C}^\mathsf{T}(\boldsymbol{Q}_t+\boldsymbol{C}\boldsymbol{x}_t+\boldsymbol{d}_{t-1})-2\alpha\boldsymbol{x}_t\right]\Big\},\end{aligned}\tag{64}$$

where $\alpha>0$ is the stepsize parameter. Since $\boldsymbol{\Lambda}_t$ is diagonal, the minimization in (64) can be distributed in each entry of $\boldsymbol{x}$. Thus, the action update can be conducted in closed-form in parallel as follows for any j,k:

$$\begin{aligned}y_{t+1,jk}=\Bigg[\frac{1}{2(\xi_{t,jk}+\alpha)}\Bigg(&Q_{t,j}-Q_{t,J+k}-\sum_{k'=1}^{K}y_{t,jk'}\\ &-\sum_{j'=1}^{J}y_{t,j'k}+z_{t,k}+d_{t-1,j}+2\alpha y_{t,jk}\Bigg)\Bigg]_0^{\bar{y}_{jk}},\end{aligned}\tag{65}$$

$$z_{t+1,k}=\left[\frac{1}{2(\eta_{t,k}+\alpha)}\left(Q_{t,J+k}+\sum_{j=1}^{J}y_{t,jk}+(2\alpha-1)z_{t,k}\right)\right]_0^{\bar{z}_k},\tag{66}$$

where $[x]_a^b=\min\{b,\max\{x,a\}\}$. Furthermore, according to (6), the virtual queues can also be updated in parallel as: $\forall j=1,\ldots,J, k=1,\ldots,K$,

$$Q_{t+1,j}=\max\left\{\sum_{k=1}^{K}y_{t+1,jk}-d_{t,j},Q_{t,j}-\sum_{k=1}^{K}y_{t+1,jk}+d_{t,j}\right\},\tag{67}$$

$$\begin{aligned}Q_{t+1,J+k}=\max\Bigg\{&-\sum_{j=1}^{J}y_{t+1,jk}+z_{t+1,k},\\ &Q_{t,J+k}+\sum_{j=1}^{J}y_{t+1,jk}-z_{t+1,k}\Bigg\}.\end{aligned}\tag{68}$$

Equations (65), (66), (67) and (68) altogether constitute a distributed implementation of Algorithm 1 for the dynamic resource allocation problem (63). In summary, for the data center resource allocation problem under study, $\{d_{t,j},\xi_{t,jk},\eta_{t,k}\}$ are sequential problem data; $\{y_{t,jk},z_{t,k}\}$ are sequential control variables; and $\{\boldsymbol{Q}_t\}$ is a sequence of auxiliary variables used to facilitate the implementation of Algorithm 1.

B. Numerical Results

In this subsection, numerical experiments are carried out to confirm the effectiveness of the Algorithm 1 for the dynamic resource allocation problem in data centers. Specifically, we consider a cloud computing network of $J = 10$ mapping nodes and $K = 10$ data centers. All the maximum data transmission rates $\{\bar{y}_{jk}\}$ are randomly and uniformly distributed in $[10, 100]$ while all the maximum data processing rates $\{\bar{z}_k\}$ are randomly and uniformly distributed in $[50, 500]$. As for the time-varying cost parameters $\{\xi_{t,jk}\}$ and $\{\eta_{t,k}\}$ and arriving rates $\{d_{t,j}\}$, we distinguish the following two models of time-variability.

- *Markovian parameters:* Let τ be some positive constant controlling the varying rate of the function sequences The bigger τ is, the slower the parameters (and thus the objective and constraint functions) vary across time. The time-varying parameters $\{d_{t,j}\}$, $\{\xi_{t,jk}\}$ and $\{\eta_{t,k}\}$ are updated in a Markovian manner as follows:

$$d_{t,j} = \left[d_{t-1,j} + t^{-\tau} e_{t,j}^{(d)}\right]^+, \ \forall j, \tag{69}$$

$$\xi_{t,jk} = \left[\xi_{t-1,jk} + t^{-\tau} e_{t,jk}^{(\xi)}\right]^+, \ \forall j, k, \tag{70}$$

$$\eta_{t,k} = \left[\eta_{t-1,k} + t^{-\tau} e_{t,k}^{(\eta)}\right]^+, \ \forall k, \tag{71}$$

  where $e_{t,j}^{(d)}$, $e_{t,jk}^{(\xi)}$ and $e_{t,k}^{(\eta)}$ are independent random variables uniformly and randomly distributed in $[-5, 5]$, $[-0.25, 0.25]$ and $[-0.5, 0.5]$, respectively. The initial values of the parameters are generated as follows. $d_{0,j}$ is uniformly distributed on $[0, 10]$; $\xi_{0,jk}$ is uniformly distributed on $[0, 0.5]$; $\eta_{0,k}$ is uniformly distributed on $[0, 1]$.
- *Noisy periodic parameters:* Let $T_0 > 0$ be the period. Define the initial phases $\phi_j^{(d)}$, $\phi_{kj}^{(\xi)}$ and $\phi_k^{(\eta)}$ to be indepedent random variables uniformly distributed in $[0, 2\pi]$. The time-varying parameters $\{d_{t,j}\}$, $\{\xi_{t,jk}\}$ and $\{\eta_{t,k}\}$ are generated as follows:

$$d_{t,j} = 10 \sin\left(\frac{2\pi t}{T_0} + \phi_j^{(d)}\right) + v_{t,j}^{(d)}, \ \forall j, \tag{72}$$

$$\xi_{t,kj} = 0.5 \sin\left(\frac{2\pi t}{T_0} + \phi_{kj}^{(\xi)}\right) + v_{t,kj}^{(\xi)}, \ \forall j, k, \tag{73}$$

$$\eta_{t,k} = \sin\left(\frac{2\pi t}{T_0} + \phi_k^{(\eta)}\right) + v_{t,k}^{(\eta)}, \ \forall k, \tag{74}$$

  where the noises $v_{t,j}^{(d)}$, $v_{t,kj}^{(\xi)}$ and $v_{t,k}^{(\eta)}$ are uniformly and randomly distributed over $[10, 11]$, $[0.5, 0.55]$ and $[1, 1.1]$, respectively.

We consider a total time horizon of $T = 500$. The stepsize parameter α of Algorithm 1 is chosen to be 10. For comparison purpose, besides Algorithm 1, we also simulate the MOSP, a state-of-the-art constrained OCO algorithm suitable for time-varying constraints [26]. The stepsize parameters of MOSP are chosen as $\rho = \mu = T^{-\frac{1}{3}} = 0.126$ according to the stepsize rule in [26] for the case of no knowledge of variations. We note that the role of $\frac{1}{\alpha}$ in Algorithm 1 is similar to that of ρ in MOSP. Thus, the stepsize choices of the two algorithms are close.

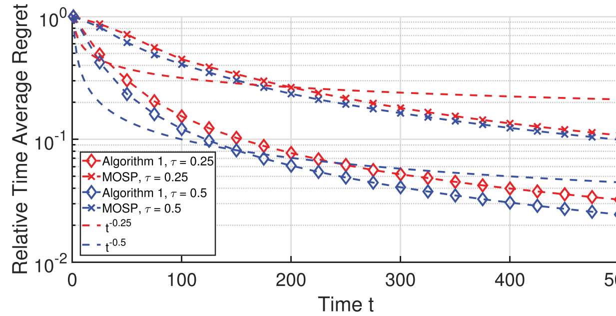


Fig. 1. Relative time average regret $\frac{\mathtt{Reg}(t)}{t \cdot \mathtt{Reg}(1)}$ versus time t with Markovian objective and constraint function sequences.

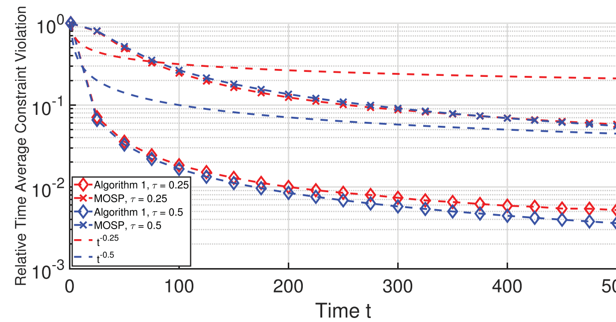


Fig. 2. Relative time average constraint violation $\frac{\left\|[\mathbf{Vio}(t)]^+\right\|_2}{t \cdot \left\|[\mathbf{Vio}(1)]^+\right\|_2}$ versus time t with Markovian objective and constraint function sequences

We first report the relative time average regret $\frac{\mathtt{Reg}(t)}{t \cdot \mathtt{Reg}(1)}$ and the relative time average constraint violation $\frac{\left\|[\mathbf{Vio}(t)]^+\right\|_2}{t \cdot \left\|[\mathbf{Vio}(1)]^+\right\|_2}$ for Markovian objective and constraint function sequences in Figs. 1 and 2, respectively, where $\mathbf{Vio}(t) = [\mathtt{Vio}_1(t), \ldots, \mathtt{Vio}_m(t)]^\mathsf{T}$. Two values of τ are considered: $\tau = 0.25$ and $\tau = 0.5$, in which the former represents faster varying rate of the function sequences than the latter does. We observe that the performance of Algorithm 1 is remarkably better than that of MOSP in terms of both regret and constraint violations in either cases of τ. This generalizes the performance adavantage of the queue inspired algorithm in [1] over dual subgradient method to the scenario of time-varying objective/constraint functions, which are unknown a priori. Furthermore, we remark that the performance of Algorithm 1 is robust to the varying rate of the function sequences. The regret and constraint violation of the case $\tau = 0.25$ is only slightly bigger than that of the case $\tau = 0.5$. In either cases of τ, the time average regret and time average constraint violation of Algorithm 1 can converge to zero, i.e., the regret and constraint violation of Algorithm 1 are sublinear, as guaranteed by Theorems 1 and 2. Additionally, from the generation process of the Markovian parameters, we know that[4] $\|f_t - f_{t+1}\|_\infty = \Theta\left(t^{-\tau}\right)$, i.e., $\|f_t - f_{t+1}\|_\infty$ is on the order of $t^{-\tau}$ for large t. Thus, according to (13), we have $V_f(t) = \Theta\left(t^{1-\tau}\right)$. Similarly, we can

[4]We follow the standard definition of $\Theta(\cdot)$. That is, for sequences a_n and b_n, $a_n = \Theta(b_n)$ means $a_n = \mathcal{O}(b_n)$ and $b_n = \mathcal{O}(a_n)$.

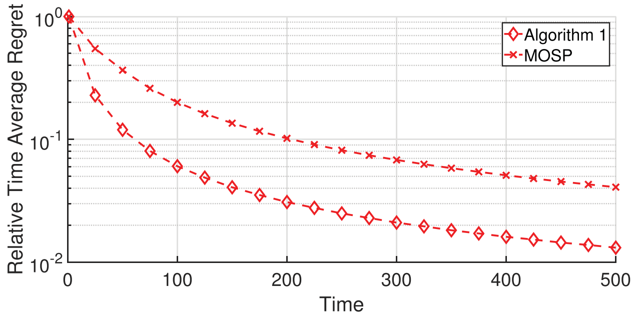


Fig. 3. Relative time average regret $\frac{\mathtt{Reg}(t)}{t \cdot \mathtt{Reg}(1)}$ versus time t with noisy periodic objective and constraint function sequences.

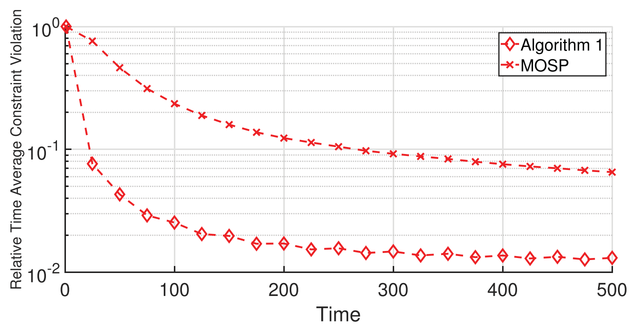


Fig. 4. Relative time average constraint violation $\frac{\|[\mathbf{Vio}(t)]^+\|_2}{t \cdot \|[\mathbf{Vio}(1)]^+\|_2}$ versus time t with noisy periodic objective and constraint function sequences.

obtain $V_{\boldsymbol{x}}(t) = \Theta(t^{1-\tau})$, $V_{\boldsymbol{g}}(t) = \Theta(t^{1-2\tau})$, $\widetilde{V}_{\boldsymbol{g}}(t) = \Theta(t^{1-\tau})$, and $V_{\boldsymbol{\lambda}}(t) = \Theta(t^{1-\tau})$. Therefore, by Theorems 1 and 2, we get the asymptotic time-average performance bounds as $\frac{\mathtt{Reg}(t)}{t} \leq \Theta(t^{-\tau})$ and $\frac{\mathtt{Vio}_k(t)}{t} \leq \Theta(t^{-\tau})$, in which constant factors are omitted. To compare the numerical performance of Algorithm 1 with the corresponding theoretical bounds, we plot $t^{-\tau}$ in Figs. 1 and 2, in which τ is 0.25 or 0.5. We observe that, for either value of τ, the relative time average regret and the relative time average constraint violation are smaller than the corresponding $t^{-\tau}$ for large enough time t, confirming the theoretical bounds. We note that the numerical relative time average regrets and constraint violations can be larger than the corresponding $t^{-\tau}$ for small t since $\Theta(t^{-\tau})$ is asymptotic bound in order sense, which omits constant factors. Moreover, we remark that the theoretical bounds for regrets and constraint violations can be quite conservative and the actual numerical performance can be much better than the corresponding theoretical bounds for large t.

The relative time average regret and the relative time average constraint violation for noisy periodic objective and constraint function sequences are shown in Figs. 3 and 4, respectively, where the period is set to be $T_0 = 50$. In such a case, we note that the variations of problem data are not sublinear so that the sufficient conditions for sublinear regret and constraint violations in Theorems 1 and 2 do not hold. In fact, the theoretical bounds of the regret and the constraint violations in Theorems 1 and 2 are linear, i.e., $\mathtt{Reg}(t) \leq \Theta(t)$ and $\mathtt{Vio}_k(t) \leq \Theta(t)$, which are not very useful. Interestingly, the relative time average regret and the relative time average constraint violation of Algorithm 1 can still converge to zero empirically, i.e., the numerical regrets and constraint violations are still sublinear for this particular simulation setup. This indicates that the theoretical bounds in Theorems 1 and 2 are not always tight and the corresponding conditions for sublinear regret and constraint violations are only sufficient conditions. Even when these sufficient conditions do not hold, e.g., when the loss/constraint functions are chosen by a non-oblivious adversary, it is still possible for Algorithm 1 to generate sublinear regrets and constraint violations, i.e., Algorithm 1 may still work well. In addition, analogous to the Markovian case, we observe that the numerical performance of Algorithm 1 is still considerably better than that of MOSP in terms of both regret and constraint violation.

V. Conclusion

In this paper, we have studied constrained OCO with time-varying objective and constraint functions. A novel online algorithm based on virtual queues has been developed. Adopting a dynamic benchmark sequence, we have established upper bounds of the regret and constraint violations of the algorithm in terms of the temporal variations of the underlying dynamic optimization problems. The algorithm possesses sublinear regret and sublinear constraint violations provided that the temporal variations of the optimization problems are sublinear, i.e., the objective and constraint functions do not vary too drastically over time. The analytical bounds of the proposed algorithm are superior to those of MOSP in [26] in most scenarios and the choice of the stepsize parameter does not rely on the total time horizon of the online optimization procedure. Finally, we have applied the algorithm to a dynamic resource allocation problem in data center networks. Numerical experiments have demonstrated the effectiveness of the algorithm and its performance improvement relative to the MOSP.

Appendix A
Proof of Lemma 1

1) Fix any $k = 1, \ldots, m$. When $t = 0$, $Q_{0,k} = 0$, so it is clearly nonnegative. We use induction to show this holds for any t. Suppose for some $t \geq 0$, $Q_{t,k} \geq 0$. If $g_{t,k}(\boldsymbol{x}_{t+1}) \geq 0$, then by (6), $Q_{t+1,k} \geq Q_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1}) \geq 0$. If $g_{t,k}(\boldsymbol{x}_{t+1}) < 0$, then $Q_{t+1,k} \geq -g_{t,k}(\boldsymbol{x}_{t+1}) > 0$. So, we always have $Q_{t+1,k} \geq 0$.
2) For $t \geq 0$, we have $Q_{t+1,k} \geq -g_{t,k}(\boldsymbol{x}_{t+1})$, i.e., $Q_{t+1,k} + g_{t,k}(\boldsymbol{x}_{t+1}) \geq 0$. When $t = -1$, the result trivially holds.
3) Fix any $k = 1, \ldots, m$. For $t \geq 0$, if $g_{t,k}(\boldsymbol{x}_{t+1}) \geq 0$, we have:

$$Q_{t+1,k} \geq Q_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1}) \overset{\text{(a)}}{\geq} g_{t,k}(\boldsymbol{x}_{t+1}) = |g_{t,k}(\boldsymbol{x}_{t+1})|, \tag{A1}$$

where (a) results from part (1) of Lemma 1. If $g_{t,k}(\boldsymbol{x}_{t+1}) < 0$, then $Q_{t+1,k} \geq -g_{t,k}(\boldsymbol{x}_{t+1}) = |g_{t,k}(\boldsymbol{x}_{t+1})|$. Thus, we always have $Q_{t+1,k} \geq |g_{t,k}(\boldsymbol{x}_{t+1})|$. So, $Q_{t+1,k}^2 \geq g_{t,k}^2(\boldsymbol{x}_{t+1})$. Summing over all k yields $\|\boldsymbol{Q}_{t+1}\|_2 \geq \|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2$.

APPENDIX B
PROOF OF LEMMA 2

For $t \geq 0$, we have $Q_{t+1,k} = Q_{t,k} + b_{t,k}$, where $b_{t,k}$ is defined as:

$$b_{t,k} := \begin{cases} -g_{t,k}(\boldsymbol{x}_{t+1}) - Q_{t,k}, & \text{if } -g_{t,k}(\boldsymbol{x}_{t+1}) \geq Q_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1}), \\ g_{t,k}(\boldsymbol{x}_{t+1}), & \text{otherwise.} \end{cases}$$

Thus, we have:

$$(b_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1}) + Q_{t,k})(b_{t,k} - g_{t,k}(\boldsymbol{x}_{t+1})) = 0. \tag{B1}$$

$$\frac{1}{2}Q_{t+1,k}^2 = \frac{1}{2}Q_{t,k}^2 + \frac{1}{2}b_{t,k}^2 + Q_{t,k}b_{t,k} \tag{B2}$$

$$= \frac{1}{2}Q_{t,k}^2 + \frac{1}{2}b_{t,k}^2 + Q_{t,k}g_{t,k}(\boldsymbol{x}_{t+1}) + Q_{t,k}(b_{t,k} - g_{t,k}(\boldsymbol{x}_{t+1})) \tag{B3}$$

$$= \frac{1}{2}Q_{t,k}^2 + \frac{1}{2}b_{t,k}^2 + Q_{t,k}g_{t,k}(\boldsymbol{x}_{t+1}) - (b_{t,k} + g_{t,k}(\boldsymbol{x}_{t+1}))(b_{t,k} - g_{t,k}(\boldsymbol{x}_{t+1})) \tag{B4}$$

$$\leq \frac{1}{2}Q_{t,k}^2 + Q_{t,k}g_{t,k}(\boldsymbol{x}_{t+1}) + g_{t,k}^2(\boldsymbol{x}_{t+1}). \tag{B5}$$

Summing (B5) over all k gives:

$$\Delta_t \leq \boldsymbol{Q}_t^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}_{t+1}) + \|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2. \tag{B6}$$

APPENDIX C
PROOF OF LEMMA 3

The proof makes use of the following fact [5], [33].

Fact 1: Let $\mathbb{S} \subset \mathbb{R}^n$ be a convex set. Let $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ be a strongly convex function with modulus $c > 0$, i.e., $\phi(\boldsymbol{x}) - \frac{c}{2}\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}$ is convex. Denote $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{S}} \phi(\boldsymbol{x})$. Then, for any $x \in \mathbb{S}$, we have $\phi(\boldsymbol{x}^*) \leq \phi(\boldsymbol{x}) - \frac{c}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2$.

By Lemma 1-(2), we know, for $t \geq 0$, $\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t) \succeq \boldsymbol{0}$. So, for $t \geq 0$, $f_t(\boldsymbol{x}) + (\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t))^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x})$ is convex with respect to $\boldsymbol{x}$. Thus, $f_t(\boldsymbol{x}) + (\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t))^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}) + \alpha\|\boldsymbol{x} - \boldsymbol{x}_t\|_2^2$ is strongly convex with modulus 2α. Note the action update in (5). Thus, by Fact 1, we thus have:

$$\begin{aligned} &f_t(\boldsymbol{x}_{t+1}) + (\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t))^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}_{t+1}) + \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|_2^2 \\ &\quad \leq f_t(\boldsymbol{x}_t^*) + (\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t))^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}_t^*) + \alpha\|\boldsymbol{x}_t^* - \boldsymbol{x}_t\|_2^2 \\ &\qquad - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2 \end{aligned} \tag{C1}$$

$$\overset{(a)}{\leq} f_t(\boldsymbol{x}_t^*) + \alpha\|\boldsymbol{x}_t^* - \boldsymbol{x}_t\|_2^2 - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2, \tag{C2}$$

where (a) is due to $\boldsymbol{Q}_t + \boldsymbol{g}_{t-1}(\boldsymbol{x}_t) \succeq \boldsymbol{0}$ and $\boldsymbol{g}_t(\boldsymbol{x}_t^*) \preceq \boldsymbol{0}$. We note:

$$\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}_{t+1}) = \frac{1}{2}\Big(\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 + \|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 - \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2\Big), \tag{C3}$$

and

$$\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2$$

$$= \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t) + \boldsymbol{g}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 \tag{C4}$$

$$\leq 2\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t)\|_2^2 + 2\|\boldsymbol{g}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 \tag{C5}$$

$$\overset{(a)}{\leq} 2\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t)\|_2^2 + 2\beta^2\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2^2, \tag{C6}$$

where in (a) we make use of the Lipschitz continuity of $\boldsymbol{g}_t$ (Assumption 3). Hence, from (C2), we have:

$$f_t(\boldsymbol{x}_{t+1}) + \boldsymbol{Q}_t^{\mathsf{T}}\boldsymbol{g}_t(\boldsymbol{x}_{t+1})$$

$$\begin{aligned} &\overset{(a)}{\leq} f_t(\boldsymbol{x}_t^*) + \alpha\|\boldsymbol{x}_t^* - \boldsymbol{x}_t\|_2^2 - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2 \\ &\quad - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|_2^2 + \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 \\ &\quad - \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 - \frac{1}{2}\|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 \end{aligned} \tag{C7}$$

$$\begin{aligned} &\overset{(b)}{\leq} f_t(\boldsymbol{x}_t^*) + \alpha\|\boldsymbol{x}_t^* - \boldsymbol{x}_t\|_2^2 - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2 \\ &\quad - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|_2^2 + \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t)\|_2^2 \\ &\quad + \beta^2\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_2^2 - \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 - \frac{1}{2}\|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2 \end{aligned} \tag{C8}$$

$$\begin{aligned} &\overset{(c)}{\leq} f_t(\boldsymbol{x}_t^*) + \alpha\|\boldsymbol{x}_t^* - \boldsymbol{x}_t\|_2^2 - \alpha\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t^*\|_2^2 \\ &\quad + \|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t)\|_2^2 \\ &\quad - \frac{1}{2}\|\boldsymbol{g}_{t-1}(\boldsymbol{x}_t)\|_2^2 - \frac{1}{2}\|\boldsymbol{g}_t(\boldsymbol{x}_{t+1})\|_2^2, \end{aligned} \tag{C9}$$

where (a) results from (C3); (b) results from (C6); and (c) results from Assumption 7. Adding (C9) with (7) yields the desired result (8).

REFERENCES

[1] H. Yu and M. J. Neely, "A simple parallel algorithm with an o(1/t) convergence rate for general convex programs," *SIAM J. Optim.*, vol. 27, no. 2, pp. 759–783, 2017.

[2] E. Hazan *et al.*, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, no. 3–4, pp. 157–325, 2016.

[3] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[5] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[6] Y. Zhang, M. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, "Peak-aware online economic dispatching for microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 323–335, Jan. 2018.

[7] L. Lu, J. Tu, C.-K. Chau, M. Chen, and X. Lin, "Online energy generation scheduling for microgrids with intermittent energy sources and co-generation," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, ACM, New York, NY, USA, 2013, pp. 53–66.

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

[9] Z. Liu, I. Liu, S. Low, and A. Wierman, "Pricing data center demand response," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 42, no. 1, pp. 111–123, 2014.

[10] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen, "Data center demand response: Avoiding the coincident peak via workload shifting and local generation," *Performance Evaluation*, vol. 70, no. 10, pp. 770–791, 2013.

[11] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 928–935.

[12] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2, pp. 169–192, 2007.

[13] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Oper. Res.*, vol. 63, no. 5, pp. 1227–1244, 2015.

[14] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Select. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[15] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, Philadelphia, PA, USA, 2005, pp. 385–394.

[16] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. Conf. Learn. Theory*, 2010, pp. 28–40.

[17] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew, "Online convex optimization using predictions," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 43, pp. 191–204, 2015.

[18] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, "Using predictions in online optimization: Looking forward with an eye on the past," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Sci.*, 2016, pp. 193–206.

[19] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, pp. 2503–2528, 2012.

[20] S. Paternain and A. Ribeiro, "Online learning of feasible strategies in unknown environments," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2807–2822, Jun. 2017.

[21] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1699–1706.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[23] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5149–5164, Oct. 2015.

[24] A. Koppel, B. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3062–3077, Jun. 2017.

[25] J. D. Abernethy, E. Hazan, and A. Rakhlin, "Interior-point methods for full-information and bandit online learning," *IEEE Trans. Inform. Theory*, vol. 58, no. 7, pp. 4164–4175, Jul. 2012.

[26] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.

[27] M. J. Neely and H. Yu, "Online convex optimization with time-varying constraints," arXiv:1702.04783, 2017.

[28] S. S. Haykin, *Adaptive Filter Theory*. India: Pearson Education, 2008.

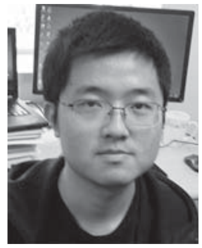
[29] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[30] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[31] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, "Stochastic averaging for constrained optimization with application to online resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3078–3093, Jun. 2017.

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

[33] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2015.



Xuanyu Cao received the B.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2013, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 2016 and 2017, respectively, all in electrical engineering. He is currently a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. His research interests include the optimization, mechanism design, game theory, signal processing, and probabilistic methods for information systems and networks.

Junshan Zhang (F'12) received the Ph.D. degree from the School of ECE, Purdue University, West Lafayette, IL, USA, in 2000. He joined the School of ECEE, Arizona State University, Tempe, AZ, USA, in August 2000, where he has been a Fulton Chair Professor since 2015. His research interests include the general field of information networks and data science, including Fog Computing and its applications in IoT and 5G, Internet of Things data privacy/security, optimization/control of mobile social networks, cognitive radio networks, stochastic modeling, and control for smart grid. Building on his research findings, he cofounded Smartiply Inc., a Fog Computing startup company delivering boosted network connectivity and embedded artificial intelligence. He was the recipient of the ONR Young Investigator Award in 2005, the NSF CAREER Award in 2003, and the IEEE Wireless Communication Technical Committee Recognition Award in 2016. His papers have won a few awards, including the Kenneth C. Sevcik Outstanding Student Paper Award of ACM SIGMETRICS/IFIP Performance 2016, the Best Paper Runner-up Award of IEEE INFOCOM 2009 and IEEE INFOCOM 2014, and the Best Paper Award at IEEE ICC 2008 and ICC 2017. He was a TPC cochair for a number of major conferences in communication networks, including IEEE INFOCOM 2012 and ACM MOBIHOC 2015. He was the general chair for ACM/IEEE SEC 2017, WiOPT 2016, and IEEE Communication Theory Workshop 2007. He was a Distinguished Lecturer of the IEEE Communications Society. He was an associate editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an editor for the *Computer Network* journal, and an editor of *IEEE Wireless Communication Magazine*. He is currently serving as an editor-at-large forIEEE/ACM TRANSACTIONS ON NETWORKING and an editor for *IEEE Network Magazine*.

H. Vincent Poor (S'72–M'77–SM'82–F'87) received the Ph.D. degree in EECS from Princeton University, Princeton, NJ, USA, in 1977. From 1977 to 1990, he was the faculty of the University of Illinois at Urbana-Champaign, Champaign, IL, USA. Since 1990, he has been on a Faculty at Princeton University, where he is the Michael Henry Strater University Professor of electrical engineering. From 2006 to 2016, he served as the Dean of the School of Engineering and Applied Science, Princeton University. He has also held visiting appointments at several other institutions, including most recently at Berkeley and Cambridge. His research interests include the areas of information theory and signal processing, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is a recent book *Information Theoretic Security and Privacy of Information Systems* (Cambridge University Press, 2017). He is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He was the recipient of the Technical Achievement and Society Awards of the IEEE Signal Processing Society in 2007 and 2011, respectively, the 2017 IEEE Alexander Graham Bell Medal, Honorary Professorships from Peking University and Tsinghua University, both conferred in 2017, and a D.Sc. honoris causa from Syracuse University, awarded in 2017.