Online Convex Optimization With Time-Varying Constraints and Bandit Feedback

Xuanyu Cao and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—In this paper, online convex optimization problem with time-varying constraints is studied from the perspective of an agent taking sequential actions. Both the objective function and the constraint functions are dynamic and unknown *a priori* to the agent. We first consider the scenario of the gradient feedback, in which, the values and gradients of the objective function and constraint functions at the chosen action are revealed after an action is submitted. We propose a computationally efficient online algorithm, which only involves direct closed-form computations at each time instant. It is shown that the algorithm possesses sublinear regret with respect to the dynamic benchmark sequence and sublinear constraint violations, as long as the drift of the benchmark sequence is sublinear, or in other words, the underlying dynamic optimization problems do not vary too drastically. Furthermore, we investigate the scenario of the bandit feedback, in which, after an action is chosen, only the values of the objective function and the constraint functions at several random points close to the action are announced to the agent. A bandit version of the online algorithm is proposed and we also establish its sublinear expected regret and sublinear expected constraint violations under the assumption that the drift of the benchmark sequence is sublinear. Finally, two numerical examples, namely online quadratic programming and online logistic regression, are presented to corroborate the effectiveness of the proposed algorithms and to confirm the theoretical guarantees.**

***Index Terms*—Bandit feedback, constrained optimization, online convex optimization (OCO), stochastic optimization.**

I. Introduction

In THE last decade, online convex optimization (OCO) has emerged as a promising paradigm and methodology for many signal processing and control problems [1], [2]. Unlike the traditional static optimization problems [3], [4], OCO is a sequential decision making procedure of an agent, who needs to choose an action at each time. The time-varying objective function and/or constraint functions are unknown to the agent *a priori*. Only after an action is chosen and submitted, (possibly partial) feedback information of the current objective/constraint functions is revealed to the agent. Due to this lack of information, it is impossible for OCO algorithms to find the exact optimal point at every time instant. Rather, a major criterion of OCO algorithms is *regret*, i.e., the performance gap between the actions induced by the algorithm and some offline optima or benchmark in hindsight. A sublinear regret is generally regarded as a good performance because it implies that, in terms of time average, the performance of the algorithm is no worse than that of the benchmark asymptotically. Such an OCO framework arises in many applications, in which, the underlying time-varying system is subject to uncertainty. Examples include smart grids with uncertain supply of renewable energy [5], [6] and data centers with uncertain user demands [7]–[9].

In [10], Zinkevich initiated the study of unconstrained OCO problems and proposed an online gradient descent algorithm, which possessed a sublinear regret of $\mathcal{O}(\sqrt{T})$ (T is the length of the time frame). The regret was further reduced to be $\mathcal{O}(\log T)$ by several online algorithms presented in [11]. While the offline benchmark was static in [10] and[11], dynamic benchmarks were adopted in [12]–[14], where algorithms with sublinear regrets were presented. In particular, improved regret bounds were developed for dynamic benchmarks in [14] under the assumption of strongly convex loss functions. In [10]–[12], each time after an action is submitted, the gradient of the objective function at the chosen action is revealed, i.e., the agent receives the *gradient feedback*. This assumption is too restrictive for many applications, in which, gradients of the objective function are hard to obtain. Instead, in these applications, only values of the objective function at the chosen action or several points near the action may be announced to the agent. This information scenario is called *bandit feedback*. For instance, after making an investment, a portfolio manager may only know the return of this particular investment choice and is unaware of the gradient of the return. Such a bandit version of the OCO problem was studied in [15] for the single-point bandit feedback and an algorithm with $\mathcal{O}(T^{\frac{3}{4}})$ regret was proposed. Later, algorithms with better regret performances were proposed in [16] for the multi-point bandit feedback. Moreover, the OCO with action switching costs and noisy predictions of objective functions was analyzed by Chen *et al.* in [17] and [18].

The aforementioned papers were concentrated on the unconstrained OCO, while many practical optimization problems involved constraints. This discrepancy motivated several works on the constrained OCO. In [19], the constrained OCO with

Manuscript received February 10, 2018; revised February 12, 2018; accepted August 24, 2018. Date of publication December 3, 2018; date of current version June 26, 2019. Recommended by Associate Editor D. Regruto. *(Corresponding author: Xuanyu Cao.)*

X. Cao is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: x.cao@princeton.edu).

K. J. Ray Liu is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: kjrliu@umd.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/TAC.2018.2884653

time-invariant constraints was studied by Mahdavi *et al.*. The OCO with affine equality constraints was investigated in [20] by using an online version of alternating direction method of multipliers (ADMM), while distributed OCO problems over networks with consensus or proximity constraints were analyzed by Koppel *et al.* in [21] and [22], respectively. In addition, an online linear optimization problem was treated under either the gradient feedback or the bandit feedback in [23]. The constraints of the OCO in all these works were time invariant and known in advance. Thus, no feedback information associated with constraints was necessary. The constrained OCO with time-varying constraints was studied by Paternain and Ribeiro in [24] for the static optimal benchmark. Furthermore, constrained OCO with time-varying constraints and dynamic benchmark sequence was studied in [25] recently. There, complete feedback information of the time-varying objective function and constraint functions was needed and a modified online saddle point algorithm was presented, which necessitated solving an optimization problem at each time instant. These limitations made the algorithm of [25] computationally inefficient and not suitable for the bandit feedback. Recently, another line of research related to OCO was a class of prediction–correction methods for time-varying optimization [26]–[28]. Different from OCO, prediction–correction methods needed information of the current objective/constraint functions to update the decision variables in the correction step, which prohibited their direct application to OCO. Besides, the implementation of prediction–correction methods were more computationally demanding than most OCO algorithms since (inverse) Hessian matrices of the objective/constraint functions were needed.

Therefore, in this paper, we are motivated to design and analyze computationally efficient algorithms for constrained OCO with time-varying constraints in the scenarios of both gradient feedback and bandit feedback. Specifically, our main contributions are summarized in the following.

1) For constrained OCO with time-varying constraints and gradient feedback, we propose a computationally efficient online algorithm (Algorithm 1), which only involves direct closed-form computations at each time instant and is amenable to a bandit setting with some modifications. We theoretically establish that Algorithm 1 achieves sublinear regret and sublinear constraint violations simultaneously as long as the drift of the dynamic benchmark sequence is sublinear, or in other words, the underlying dynamic optimization problem does not vary too drastically across time (Theorem 1). In such a case, both the time average regret and the time average constraint violations are asymptotically nonpositive.
2) For constrained OCO with time-varying constraints and bandit feedback, we propose an online algorithm (Algorithm 2) based on appropriate approximations and modifications of Algorithm 1. Sublinear expected regret and sublinear expected constraint violations are also demonstrated under the assumption of the sublinear drift of the benchmark sequence (Theorem 2). In other words, the time average expected regret and the time average expected constraint violations are asymptotically nonpositive.
3) Two numerical examples, namely online quadratic programming (OQP) and online logistic regression (OLR) are presented to corroborate the effectiveness of the proposed algorithms. We observe that, in both examples, as time progresses, the time average regrets converge to zero and the time average constraint violations become negative under both gradient feedback and bandit feedback. This confirms the theoretical guarantees in Theorems 1 and 2.

The rest of this paper is organized as follows. In Section II, constrained OCO with time-varying constraints is formally formulated for both gradient feedback and bandit feedback. In Sections III and IV, we propose and analyze algorithms for scenarios of gradient feedback and bandit feedback, respectively. Numerical experiments are presented in Section V, following which we conclude this study in Section VI.

II. Problem Formulation

In this section, we formulate OCO problems with time-varying constraints. Based on different form of feedback information, we consider two scenarios: gradient feedback and bandit feedback. The performance metrics in terms of objective function values and constraint violations as well as the pertinent assumptions and preliminaries are also presented.

A. Gradient Feedback

The classical unconstrained OCO problem can be described as the following iterative procedure between an agent and the nature [10]. Assume that time is discrete. At each time slot t, the agent selects an action $\mathbf{x}_t \in \mathbb{R}^n$ from the action set $\mathcal{X} \subset \mathbb{R}^n$. After the action $\mathbf{x}_t$ is chosen, the nature announces the gradient of the loss function $f_t : \mathbb{R}^n \mapsto \mathbb{R}$ at $\mathbf{x}_t$, i.e., $\nabla f_t(\mathbf{x}_t)$, to the agent, who experiences a loss of $f_t(\mathbf{x}_t)$ at time slot t. Such a scenario is called the *gradient feedback* as the agent obtains gradient information about the loss function after the action is chosen. The goal of the agent is to minimize the total loss over certain time frame. The action set $\mathcal{X}$ is known in advance to the agent before the OCO procedure starts and remains unchanged as the procedure progresses.

This classical OCO formulation, though being useful in many situations, cannot deal with problems with constraints [19] and especially time-varying constraints [24], [25], which naturally arise in many practical applications. For instance, in smart grids, with the high penetration of renewable energy such as solar power and wind power, the energy supply can be very uncertain and hard to predict. Thus, the controller of the power grid often needs to schedule the electricity power with time-varying and uncertain power supply in a real time manner, which can be posed as online optimization problems with time-varying resource constraints [5], [6]. Therefore, in this paper, we are motivated to study constrained OCO problem with time-varying constraints. Specifically, at each time t, after the agent chooses an action $\mathbf{x}_t \in \mathcal{X}$, the nature will announce not only $\nabla f_t(\mathbf{x}_t)$, but also the value and gradients of a vector-valued constraint function $\mathbf{g}_t : \mathbb{R}^n \mapsto \mathbb{R}^m$ at $\mathbf{x}_t$, i.e., $\mathbf{g}_t(\mathbf{x}_t)$ and $\nabla \mathbf{g}_t(\mathbf{x}_t)$ (Jacobian matrix), to the agent. The agent wants to minimize the loss $f_t(\mathbf{x}_t)$ while satisfying the time-varying constraints $\mathbf{g}_t(\mathbf{x}_t) \preceq \mathbf{0}$,

which is equivalent to the computation of $\mathbf{x}_t^*$ defined as follows:

$$\mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}}\{f_t(\mathbf{x})|\mathbf{g}_t(\mathbf{x}) \preceq \mathbf{0}\}. \tag{1}$$

However, solving problem (1) directly to choose action $\mathbf{x}_t$ is impossible in the online setting, here, as the loss function $f_t(\cdot)$ and constraint function $\mathbf{g}_t(\cdot)$ are unknown before the action $\mathbf{x}_t$ is chosen. In particular, since $\mathbf{g}_t(\cdot)$ is unknown *a priori*, the constraint $\mathbf{g}_t(\mathbf{x}_t) \preceq \mathbf{0}$ is hard to be satisfied in every time slot t. Rather, the agent tries to satisfy the constraints in the long run. In other words, the agent wants to ensure the long-term constraint of $\sum_{t=1}^T \mathbf{g}_t(\mathbf{x}_t) \preceq \mathbf{0}$ over some given period of length T. This type of long-term constraint is appropriate in many applications. For example, in a smart grid with renewable energy sources, the grid controller wants to balance the power demand by the renewable energy supply. Due to the uncertainty of renewable energy, the controller cannot depend on it solely. Instead, the controller needs to reserve some traditional energy (e.g., coal and gas) to balance the temporary deficit of the power supply from renewable energy. 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 can be 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 should be regarded as successful in operating a smart grid powered by renewable energy.

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

$$\begin{aligned}\text{Minimize}_{\mathbf{x}_1,\ldots,\mathbf{x}_T\in\mathcal{X}} & \sum_{t=1}^T f_t(\mathbf{x}_t)\\ \text{s.t.} & \sum_{t=1}^T \mathbf{g}_t(\mathbf{x}_t) \preceq \mathbf{0}.\end{aligned} \tag{2}$$

Solving problem (2) exactly is still impossible in the online setting, here, as the information about the loss functions and constraint functions is unknown *a priori*. Instead, our goal is to obtain a total loss $\sum_{t=1}^T f_t(\mathbf{x}_t)$ that is not too large compared to some benchmark and meanwhile, to ensure that $\sum_{t=1}^T \mathbf{g}_t(\mathbf{x}_t)$ is not too positive, i.e., the long-term constraint is not violated too much. As our original goal is to select the action $\mathbf{x}_t$ according to the solution of the problem (1), we choose $\{\mathbf{x}_t^*\}_{t=1}^T$ as the benchmark sequence and the first performance criterion is the regret with respect to the benchmark, which is defined as $\texttt{Reg}(T) := \sum_{t=1}^T [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]$. The second performance metric is the constraint violation $\texttt{Vio}^i(T) := \sum_{t=1}^T g_t^i(\mathbf{x}_t)$, $i = 1,\ldots,m$, where $g_t^i(\cdot)$ is the ith component of the vector-valued constraint function $\mathbf{g}_t(\cdot)$, i.e., $\mathbf{g}_t(\mathbf{x}) = [g_t^1(\mathbf{x}),\ldots,g_t^m(\mathbf{x})]^\mathsf{T}$. We note that the aforementioned definitions of regret and constraint violations are prevalent and widely accepted in the literature of OCO [1], [2]. An ideal action sequence should possess both small regret and small constraint violations. More precisely, the regret and the constraint violations should be sublinear with respect to T, i.e., $\texttt{Reg}(T) \le o(T)$ and $\texttt{Vio}^i(T) \le o(T)$ $\forall\, i = 1,\ldots,m$. Hence, as T goes to infinity, $\frac{\texttt{Reg}(T)}{T} \le o(1) \to 0$ and $\frac{\texttt{Vio}^i(T)}{T} \le o(1) \to 0$. This means that, as the time length T goes to infinity, the time-average regret $\frac{\texttt{Reg}(T)}{T}$ and the time-average constraint violation $\frac{\texttt{Vio}^i(T)}{T}$ either converge to zero or converge to some negative numbers so that the performance of the sequence $\{\mathbf{x}_t\}$ is no worse than that of the benchmark sequence $\{\mathbf{x}_t^*\}$ in terms of asymptotic time average. Furthermore, we note that another possible pair of definitions of regret and constraint violations are $\widetilde{\texttt{Reg}}(T) = \sum_{t=1}^T [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]^+$ and $\widetilde{\texttt{Vio}}^i(T) = \sum_{t=1}^T [g_t^i(\mathbf{x}_t)]^+$ so that negative individual terms do not contribute to the sum any more, where $y^+ = \max\{y, 0\}$. Nevertheless, these definitions of regret and constraint violations are hard to handle and seldom used in the literature of OCO. The corresponding sublinearity is difficult, if possible, to guarantee. Thus, we do not consider these definitions in this paper.

A similar constrained OCO problem with time-varying constraints has been investigated in [25] recently. There, to facilitate performance analysis, the authors propose a modified online saddle point (MOSP) algorithm in which the primal update is not an exact gradient descent of the Lagrangian. Instead, at each time t, the primal update needs to solve an optimization problem associated with the time-varying constraint function $\mathbf{g}_{t-1}(\cdot)$. This can be unfavorable due to the following two reasons. First, solving a nonlinear optimization problem (which generally does not admit closed-form solution) at every time instant is computationally demanding, especially, for devices with low computational capability such as the cheap sensors massively deployed in sensor networks. Second, at time t, to perform update, the agent needs the complete information about the function $\mathbf{g}_{t-1}(\cdot)$ rather than its gradient at a particular point, rendering the approach not amenable to a bandit version of the problem [15], which we will discuss more in the next subsection. We are, thus, aimed at proposing a computationally efficient online algorithm for the constrained OCO problem and analyzing its performance in terms of regret and constraint violations, which are detailed in Section III. Later in Section IV, we demonstrate that, with some approximations, the proposed algorithm can be modified to accommodate the scenario of the bandit feedback.

B. Bandit Feedback

In the previous subsection, we formulate the constrained OCO with the gradient feedback, i.e., values and gradients of the loss function $f_t(\cdot)$ and the constraint function $\mathbf{g}_t(\cdot)$ at $\mathbf{x}_t$ are revealed to the agent after the action $\mathbf{x}_t$ is chosen. However, in many practical applications, even after $\mathbf{x}_t$ is chosen, the agent still cannot access the gradients of the functions $f_t(\cdot)$ and $\mathbf{g}_t(\cdot)$. Instead, the agent only knows the values of $f_t(\cdot)$ and $\mathbf{g}_t(\cdot)$ at the particular point $\mathbf{x}_t$ or several points close to $\mathbf{x}_t$. Such an information feedback scenario is called *bandit feedback*, which has broad applications. For instance, consider the portfolio management problem with uncertain return. At time t, after the manager makes an investment decision $\mathbf{x}_t$, the nature (e.g., the stock market) will decide the loss (or negative profit) function $f_t(\cdot)$ and the manager will incur a loss of $f_t(\mathbf{x}_t)$. Afterwards, the manager may only know the incurred loss $f_t(\mathbf{x}_t)$ or the values of the loss function $f_t(\cdot)$ at several points close to $\mathbf{x}_t$ (based on the incurred

loss $f_t(\mathbf{x}_t)$, the manager or expert may be able to predict the loss if certain small changes to the investment $\mathbf{x}_t$ are made, e.g., increasing slightly the investment of the stocks of a company with stable behaviors recently). Nevertheless, the manager may not know the gradient $\nabla f_t(\mathbf{x}_t)$ accurately because she does not have access to *all* the evaluations of $f_t(\cdot)$ in the neighborhood of $\mathbf{x}_t$. Similar arguments hold for the constraint function $\mathbf{g}_t(\cdot)$.

One challenge of the bandit feedback is that the agent cannot evaluate the gradients of the loss function and the constraint functions directly. Instead, the agent has to use some stochastic approximation to substitute these gradients [15], [16]. As such, the action sequence $\{\mathbf{x}_t\}$ involves randomness and the corresponding definitions of regret and constraint violations are altered to be their expected version: $\texttt{Reg}(T) := \mathbb{E}[\sum_{t=1}^T f_t(\mathbf{x}_t)] - \sum_{t=1}^T f_t(\mathbf{x}_t^*)$ and $\texttt{Vio}^i(T) := \mathbb{E}[\sum_{t=1}^T g_t^i(\mathbf{x}_t)]$. In Section IV, we will propose an online algorithm for the constrained OCO problem with the bandit feedback and show that this algorithm achieves sublinear regret and constraint violations.

C. Assumptions and Preliminaries

To facilitate later performance analysis, we make the following technical assumptions, all of which are standard in the literature of OCO [2]. Denote the unit ball in $\mathbb{R}^n$ as $\mathbb{B}$ and the unit sphere in $\mathbb{R}^n$ as $\mathbb{S}$, i.e., $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_2 \leq 1\}$ and $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_2 = 1\}$, where $\|\cdot\|_2$ is the l_2 norm.

Assumption 1: The action set $\mathcal{X}$ is a closed convex set.

Assumption 2: There exists two positive constants R and r such that $r\mathbb{B} \subset \mathcal{X} \subset R\mathbb{B}$.

Assumption 3: The loss function f_t and constraint function g_t^i are convex for any $i = 1, \ldots, m$ and $t = 1, 2, \ldots$.

Assumption 4: All loss functions f_t and constraint functions g_t^i have uniformly bounded gradients, i.e., there exists some positive constant G such that $\|\nabla f_t(\mathbf{x})\|_2 \leq G$ and $\|\nabla g_t^i(\mathbf{x})\|_2 \leq G$ for any $\mathbf{x} \in \mathcal{X}$, $i = 1, \ldots, m$, $t = 1, 2, \ldots$.

Assumption 5: All constraint functions $\mathbf{g}_t$ are uniformly bounded, i.e., there exists some positive constant D such that $\|\mathbf{g}_t(\mathbf{x})\|_2 \leq D$ for any $\mathbf{x} \in \mathcal{X}$, $t = 1, 2, \ldots$.

Assumption 6: All loss functions f_t have uniformly bounded difference, i.e., there exists some positive constant F such that $|f_t(\mathbf{x}) - f_t(\mathbf{x}')| \leq F$ for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and $t = 1, 2, \ldots$.

We note that the constant bounds $G, D,$ and F in Assumptions 4–6 hold *uniformly* for all t. These uniform bounds cannot be implied by simply imposing continuity or continuous differentiability conditions on each individual function f_t and $\mathbf{g}_t$. Furthermore, Assumption 5 cannot follow from Assumptions 2–4. Consider a couterexample where $n = m = 1$ (all variables are scalar), $\mathcal{X} = [-1, 1]$, and $g_t(x) = t + x$ for all $x \in \mathcal{X}, t = 1, 2, \ldots$. We do not construct the loss functions $\{f_t\}$ since Assumption 5 is only related to the constraint functions $\{g_t\}$. Clearly, Assumptions 2 and 3 hold. Since $g_t'(x) = 1$ for any $x \in \mathcal{X}, t = 1, 2, \ldots$, Assumption 4 also holds. However, Assumption 5 does *not* hold because $\sup_{x \in \mathcal{X}} |g_t(x)| = t + 1 \to \infty$, as $t \to \infty$. In other words, there is no *uniform* bound for the function sequence $\{g_t\}$.

Additionally, the assumption that the origin is contained in $\mathcal{X}$ (c.f., Assumption 2) can be made without loss of generality as we can always translate $\mathcal{X}$. In essence, we only require that the action set $\mathcal{X}$ contains an interior point. The assumption that the origin is contained in $\mathcal{X}$ is for ease of exposition. In fact, this assumption is made in most works on OCO with the bandit feedback, e.g., [15] and [16], to avoid cluttered notations. Next, we define the projection operator as follows, which will be used frequently later.

Definition 1: Suppose $\mathcal{S}$ is some closed convex set in $\mathbb{R}^n$. Then, for any $\mathbf{y} \in \mathbb{R}^n$, the optimization problem $\arg\min_{\mathbf{x} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|_2$ has unique minimizer, which is called the projection of $\mathbf{y}$ onto the set $\mathcal{S}$ and is denoted as $\Pi_{\mathcal{S}}(\mathbf{y})$.

Thus, according to Assumption 1, $\Pi_{\mathcal{X}}(\cdot)$ is a well-defined projection operator. We further notice the following property of the projection operator [29], which is useful in the performance analysis.

Lemma 1: Suppose $\mathcal{S} \subset \mathbb{R}^n$ is a closed convex set and $\Pi_{\mathcal{S}}(\cdot)$ is the associated projection operator. Then, for any $\mathbf{x} \in \mathcal{S}$ and $\mathbf{y} \in \mathbb{R}^n$, we have

$$\|\mathbf{x} - \Pi_{\mathcal{S}}(\mathbf{y})\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2. \tag{3}$$

III. Constrained OCO With Gradient Feedback

In this section, we develop an online algorithm for the constrained OCO problem with gradient feedback, i.e., $\nabla f_t(\mathbf{x}_t)$, $\nabla \mathbf{g}_t(\mathbf{x}_t), \mathbf{g}_t(\mathbf{x}_t)$ are revealed to the agent after the action $\mathbf{x}_t$ is chosen. The algorithm is computationally efficient as the update at each time only involves direct closed-form computations. The algorithm is also amenable to a bandit version of the problem, which will be detailed in Section IV. Performance analysis of the algorithm is presented, indicating that the algorithm can achieve sublinear regret and sublinear constraint violations simultaneously.

A. Algorithm Development

Recall the per-slot optimization problem (1). Define the modified Lagrangian of (1) to be

$$\mathfrak{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = f_t(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{g}_t(\mathbf{x}) - \frac{\delta \eta}{2} \|\boldsymbol{\lambda}\|_2^2 \tag{4}$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier; $\eta > 0$ is the stepsize of the algorithm to be used later; and δ is some positive number to be determined by later analysis. The difference between the modified Lagrangian in (4) and the classical Lagrangian is the last term of (4), which is added to prevent $\boldsymbol{\lambda}$ from becoming too large. The proposed algorithm is an online saddle point algorithm associated with the modified Lagrangian $\mathfrak{L}_t$. Specifically, the algorithm maintains and updates the primal variable $\mathbf{x}_t$ and the dual variable $\boldsymbol{\lambda}_t$ as follows. After $\mathbf{x}_t, \boldsymbol{\lambda}_t$ are chosen, the nature reveals the loss function f_t and the constraint function $\mathbf{g}_t$ to the agent. Then, the agent will perform a primal descent step for the modified Lagrangian $\mathfrak{L}_t$ to obtain the new primal variable (i.e., the new action) $\mathbf{x}_{t+1}$ as

$$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta \nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)) \tag{5}$$

$$= \Pi_{\mathcal{X}}\left(\mathbf{x}_t - \eta\left(\nabla f_t(\mathbf{x}_t) + \sum_{i=1}^m \lambda_t^i \nabla g_t^i(\mathbf{x}_t)\right)\right). \tag{6}$$

Algorithm 1: The Algorithm for Constrained OCO With Gradient Feedback.

1: Initialize $\mathbf{x}_1 \in \mathcal{X}$ and $\boldsymbol{\lambda}_1 = \mathbf{0}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Submit the action $\mathbf{x}_t$.
4: Receive the loss function $f_t(\cdot)$ and the constraint function $g_t(\cdot)$.
5: Update the primal variable, i.e., the action, according to (6) to obtain the new action $\mathbf{x}_{t+1}$.
6: Update the dual variable according to (8) to obtain the new dual variable $\boldsymbol{\lambda}_{t+1}$.
7: **end for**

The projection onto $\mathcal{X}$ is to ensure that $\mathbf{x}_{t+1}$ is a proper action in the action set. In addition, the agent performs a dual ascent step for $\mathfrak{L}_t$ to compute the new dual variable $\boldsymbol{\lambda}_{t+1}$ as

$$\boldsymbol{\lambda}_{t+1} = \Pi_{\mathbb{R}_+^m}\left(\boldsymbol{\lambda}_t + \eta \nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\right) \tag{7}$$

$$= \Pi_{\mathbb{R}_+^m}\left(\boldsymbol{\lambda}_t + \eta\left(\mathbf{g}_t(\mathbf{x}_t) - \delta\eta\boldsymbol{\lambda}_t\right)\right) \tag{8}$$

where $\mathbb{R}_+^m$ denotes the nonnegative orthant, i.e., $\mathbb{R}_+^m = \{\mathbf{x} \in \mathbb{R}^m | \mathbf{x} \succeq \mathbf{0}\}$. The projection onto the nonnegative orthant is to make sure that the dual variable is always dual feasible. Based on the updates specified in (6) and (8), we summarize the proposed algorithm for the constrained OCO with gradient feedback in Algorithm 1. We note that the updates (6) and (8) only involve closed-form computations and do not need to solve any optimization problems, indicating a high computational efficiency of Algorithm 1.

B. Performance Analysis

Now, we proceed to analyze the performance of Algorithm 1 and show that it can achieve sublinear regret and constraint violations as long as the drift (to be defined in Lemma 3) of the benchmark sequence $\{\mathbf{x}_t^*\}$ is sublinear. We first present a lemma on the evolution of the modified Lagrangian.

Lemma 2: For any $\mathbf{x} \in \mathcal{X}$, $\boldsymbol{\lambda} \succeq \mathbf{0}$, $t = 1, 2, ...$, we have

$$\begin{aligned}&\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \mathfrak{L}_t(\mathbf{x}, \boldsymbol{\lambda}_t)\\ &\leq \frac{1}{2\eta}(\|\mathbf{x} - \mathbf{x}_t\|_2^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2 + \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t\|_2^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t+1}\|_2^2)\\ &\quad + \frac{\eta}{2}\left(\|\nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 + \|\nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2\right).\end{aligned} \tag{9}$$

■

Proof: The proof is presented in Appendix A.

Before proceeding, we give a definition of the *drift* of the benchmark sequence.

Definition 2: Define $\Delta(T) := \sum_{t=2}^T \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2$ to be the *drift* of the benchmark sequence $\{\mathbf{x}_t^*\}_{t=1}^T$.

Based on Lemma 2, making use of the notion of drift, we can further bound the gradients of the modified Lagrangian $\mathfrak{L}_t$ to obtain the following result.

Lemma 3: For any $\boldsymbol{\lambda} \succeq \mathbf{0}$, we have

$$\begin{aligned}&\sum_{t=1}^T [\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \mathfrak{L}_t(\mathbf{x}_t^*, \boldsymbol{\lambda}_t)]\\ &\leq \frac{1}{2\eta}\left(5R^2 + 2R\Delta(T) + \|\boldsymbol{\lambda}\|_2^2\right) + \frac{\eta T}{2}\left[(m+1)G^2 + 2D^2\right]\\ &\quad + \frac{\eta}{2}\left[(1+m)G^2 + 2\delta^2\eta^2\right]\sum_{t=1}^T \|\boldsymbol{\lambda}_t\|_2^2.\end{aligned} \tag{10}$$

Proof: The proof is given in Appendix B. ■

We are now ready to show that the action sequence generated by Algorithm 1 possesses sublinear regret and constraint violations provided that the drift of the benchmark sequence $\{\mathbf{x}_t^*\}$ is sublinear.

Theorem 1: Suppose the drift sequence $\{\Delta(T')\}_{T'=1}^{\infty}$ is sublinear, i.e., $\lim_{T'\to\infty} \frac{\Delta(T')}{T'} = 0$. Assume T is large enough such that $\frac{\Delta(T)}{T} \leq \frac{1}{2((m+1)G^2+1)^2}$. Set $\eta = \sqrt{\frac{\Delta(T)}{T}}$ and $\delta = (m+1)G^2 + 1$. Then, we have

$$\begin{aligned}&\sum_{t=1}^T [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]\\ &\leq \frac{5R^2}{2}\sqrt{\frac{T}{\Delta(T)}} + \left(R + \frac{m+1}{2}G^2 + D^2\right)\sqrt{T\Delta(T)}\\ &= \mathcal{O}\left(\sqrt{T\Delta(T)}\right)\end{aligned} \tag{11}$$

and for any $i = 1, \ldots, m$

$$\begin{aligned}&\sum_{t=1}^T g_t^i(\mathbf{x}_t)\\ &\leq \sqrt{2\left(((m+1)G^2+1)\sqrt{T\Delta(T)} + \sqrt{\frac{T}{\Delta(T)}}\right)}\\ &\quad \times \sqrt{FT + \frac{5R^2}{2}\sqrt{\frac{T}{\Delta(T)}} + \left(R + \frac{m+1}{2}G^2 + D^2\right)\sqrt{T\Delta(T)}}\end{aligned} \tag{12}$$

$$= \mathcal{O}\left(T^{\frac{3}{4}}\Delta(T)^{\frac{1}{4}}\right). \tag{13}$$

Proof: Substituting the definition of the modified Lagrangian $\mathfrak{L}_t$ in (4) into (10) in Lemma 3 and rearranging terms, we have, for any $\boldsymbol{\lambda} \succeq \mathbf{0}$

$$\begin{aligned}&\sum_{t=1}^T [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)] + \sum_{i=1}^m \sum_{t=1}^T \left[\lambda^i g_t^i(\mathbf{x}_t) - \lambda_t^i g_t^i(\mathbf{x}_t^*)\right]\\ &\quad - \frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|_2^2\\ &\leq \frac{\eta}{2}\left[(1+m)G^2 + 2\delta^2\eta^2 - \delta\right]\sum_{t=1}^T \|\boldsymbol{\lambda}_t\|_2^2\\ &\quad + \frac{1}{2\eta}\left(5R^2 + 2R\Delta(T) + \|\boldsymbol{\lambda}\|_2^2\right)\end{aligned} \tag{14}$$

$$+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right] \tag{15}$$

$$\leq \frac{1}{2\eta}\left(5R^2+2R\Delta(T)+\|\boldsymbol{\lambda}\|_2^2\right)+\frac{\eta T}{2}[(m+1)G^2+2D^2]. \tag{16}$$

The last step (16) results from the fact $(1+m)G^2+2\delta^2\eta^2-\delta\leq 0$. This is true due to the choice of $\delta=(m+1)G^2+1$ and that T is large enough such that $\frac{\Delta(T)}{T}\leq\frac{1}{2((m+1)G^2+1)^2}$. Rearranging terms in (14) and (16), we obtain

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^*)]$$

$$+\sum_{i=1}^{m}\left[\lambda^i\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)-\left(\frac{\delta\eta T}{2}+\frac{1}{2\eta}\right)\left(\lambda^i\right)^2\right] \tag{17}$$

$$\leq\sum_{i=1}^{m}\sum_{t=1}^{T}\lambda_t^i g_t^i(\mathbf{x}_t^*)+\frac{1}{2\eta}\left(5R^2+2R\Delta(T)\right)$$

$$+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right] \tag{18}$$

$$\leq\frac{1}{2\eta}\left(5R^2+2R\Delta(T)\right)+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right] \tag{19}$$

where the last step is due to $\mathbf{g}_t(\mathbf{x}_t^*)\preceq\mathbf{0}$ and $\boldsymbol{\lambda}_t\succeq\mathbf{0}$. Note that the relation between (17) and (19) holds for any $\boldsymbol{\lambda}\succeq\mathbf{0}$. Define $x^+:=\max\{x,0\}$ for $x\in\mathbb{R}$. Maximizing the second term of (17) over $\boldsymbol{\lambda}\succeq\mathbf{0}$, i.e., choosing $\lambda_i=\frac{[\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)]^+}{\delta\eta T+\frac{1}{\eta}}$, $i=1,\ldots,m$, we obtain

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^*)]+\sum_{i=1}^{m}\frac{\left(\left[\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)\right]^+\right)^2}{2(\delta\eta T+\frac{1}{\eta})}$$

$$\leq\frac{1}{2\eta}\left(5R^2+2R\Delta(T)\right)+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right]. \tag{20}$$

Hence

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^*)]$$

$$\leq\frac{1}{2\eta}\left(5R^2+2R\Delta(T)\right)+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right]. \tag{21}$$

Substituting the choice $\eta=\sqrt{\frac{\Delta(T)}{T}}$ into (21) yields the desired result of the regret in (11). In addition, according to Assumption 6, we have $f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^*)\geq -F$. Therefore, for any $i=1,\ldots,m$

$$-FT+\frac{\left(\left[\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)\right]^+\right)^2}{2(\delta\eta T+\frac{1}{\eta})}$$

$$\leq\frac{1}{2\eta}\left(5R^2+2R\Delta(T)\right)+\frac{\eta T}{2}\left[(m+1)G^2+2D^2\right]. \tag{22}$$

Hence, we have

$$\sum_{t=1}^{T}g_t^i(\mathbf{x}_t) \tag{23}$$

$$\leq\left[\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)\right]^+ \tag{24}$$

$$\leq\sqrt{2\left(FT+\frac{1}{2\eta}(5R^2+2R\Delta(T))+\frac{\eta T}{2}[(m+1)G^2+2D^2]\right)}$$

$$\times\sqrt{\left(\delta\eta T+\frac{1}{\eta}\right)}. \tag{25}$$

Substituting $\eta=\sqrt{\frac{\Delta(T)}{T}}$ into (25) yields the desired result of the constraint violations in (12) and (13). ■

The last equality of (11) follows from the fact that the cumulative drift $\Delta(T)$ is at least $\Omega(1)$, i.e., it is no smaller than constant in order sense. In addition, as T goes to infinity, the stepsize η converges to zero in the limit since $\Delta(T)$ is sublinear. This diminishing stepsize is common in the literature of optimization [3], [4]. Furthermore, two remarks are given as follows.

Remark 1: Since the drift sequence $\Delta(\cdot)$ is sublinear, (11) and (13) are both sublinear and so are the regrets $\texttt{Reg}(T)=\sum_{t=1}^{T}[f_t(\mathbf{x}_t)-f_t(\mathbf{x}_t^*)]$ and the constraint violations $\texttt{Vio}^i(T)=\sum_{t=1}^{T}g_t^i(\mathbf{x}_t)$, $i=1,\ldots,m$. The hypothesis that the drift $\Delta(\cdot)$ is sublinear is reasonable in order to guarantee sublinear regret and constraint violations of any online algorithm. Otherwise, the drift is linear or superlinear, indicating that the benchmark sequence $\{\mathbf{x}_t^*\}$ evolves at a constant speed at least. Note that when determining action $\mathbf{x}_t$, an online algorithm does not know f_t and $\mathbf{g}_t$, which are necessary to compute $\mathbf{x}_t^*$ in (1). An online algorithm has to estimate $\{\mathbf{x}_t^*\}$ based on the available information about $f_1,\ldots,f_{t-1}$ and $\mathbf{g}_1,\ldots,\mathbf{g}_{t-1}$, which is not very helpful because $\mathbf{x}_t^*$ deviates from the past significantly (at a constant speed at least). As such, the online algorithm cannot track the benchmark sequence $\{\mathbf{x}_t^*\}$ well or more precisely, the performance metrics, i.e., the regret $\texttt{Reg}(T)$ and the constraint violations $\texttt{Vio}^i(T)$ may not be sublinear. Furthermore, we note that the prediction–correction methods in [26]–[28] do not need assumptions on the drift of the benchmark to guarantee convergence. The main reason is that, when computing the new variables $\mathbf{x}_{t+1}$, the prediction–correction methods make use of the new objective/constraint functions f_{t+1} and $\mathbf{g}_{t+1}$ in the correction step. In contrast, in the proposed Algorithm 1, to compute $\mathbf{x}_{t+1}$ and $\boldsymbol{\lambda}_{t+1}$, we only need information about the past objective/constraint functions f_t and $\mathbf{g}_t$ (c.f. (6) and (8)). This information advantage of prediction–correction methods renders them easier in tracking the benchmark so that no assumption on the drift of the benchmark is necessary to ensure convergence. Nevertheless, these methods are infeasible for OCO of this paper, in which (partial) information about f_{t+1} and $\mathbf{g}_{t+1}$ is revealed only after $\mathbf{x}_{t+1}$ is determined.

Remark 2: As discussed in Section II-A, the sublinearity of the regret $\texttt{Reg}(T)$ and the constraint violations $\texttt{Vio}^i(T)$ guaranteed in Theorem 1 implies that the time average regret $\frac{\texttt{Reg}(T)}{T}$

and the time average constraint violations $\frac{\texttt{Vio}^i(T)}{T}$ are asymptotically non-positive as the length of the time frame T goes to infinity. This suggests that, asymptotically, the action sequence generated by Algorithm 1 has performance no worse than the dynamic benchmark $\{\mathbf{x}_t^*\}$ does.

In Theorem 1, the stepsize is chosen as $\eta = \sqrt{\frac{\Delta(T)}{T}}$, which depends on the drift $\Delta(T)$. We note that $\Delta(T)$ may not be known to the agent precisely. Nevertheless, as long as $\eta = \Theta(\sqrt{\frac{\Delta(T)}{T}})$, i.e., η has the same order as $\sqrt{\frac{\Delta(T)}{T}}$ does, the order bounds of the regret and the constraint violations in Theorem 1 will hold. So, when selecting the stepsize, we only need an estimate of the *order* of $\Delta(T)$ with a possible constant factor error to ensure sublinear regret and constraint violations. Furthermore, even if such an estimate of the order of the drift $\Delta(T)$ is not available, we can still choose stepsize η to ensure sublinearity of regrets and constraint violations, as specified in the following. We presume that a sublinear upper bound of $\Delta(T)$ is known. That is, we know a sublinear positive sequence $\widetilde{\Delta}(T)$, i.e., $\lim_{T\to\infty} \frac{\widetilde{\Delta}(T)}{T} = 0$, such that $\Delta(T) \leq \widetilde{\Delta}(T)$ for T large enough (there can be a positive constant factor on either side of the inequality, which does not affect statements in order sense). Since $\Delta(T)$ is sublinear, such a sublinear upper bound $\widetilde{\Delta}(T)$ must exist and can be known in advance in many scenarios. If the agent has little knowledge about $\Delta(T)$ besides sublinearity, she can choose a very conservative sublinear upper bound $\widetilde{\Delta}(T)$, e.g., T^ζ, where $\zeta < 1$ is very close to 1. With such a sublinear upper bound $\widetilde{\Delta}(T)$ known prior to the start of the OCO, we can choose the stepsize to be $\eta = \sqrt{\frac{\widetilde{\Delta}(T)}{T}}$. Then, after minor adaption of the proof of Theorem 1, the regret bound becomes $O(\sqrt{T\widetilde{\Delta}(T)})$, while the constraint violation bound becomes $O(T^{\frac{3}{4}}\widetilde{\Delta}(T)^{\frac{1}{4}})$. Since $\widetilde{\Delta}(T)$ is sublinear, so are the regret and constraint violations. We summarize the aforementioned points in the following corollary.

Corollary 1: Suppose the drift sequence $\Delta(T)$ is sublinear. Furthermore, presume that, before the start of the OCO, the agent knows a sublinear positive sequence $\widetilde{\Delta}(T)$ such that $\Delta(T) \leq \widetilde{\Delta}(T)$ for T large enough. Set $\eta = \sqrt{\frac{\widetilde{\Delta}(T)}{T}}$ and $\delta = (m+1)G^2 + 1$. Then, we have

$$\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)] \leq \mathcal{O}\left(\sqrt{T\widetilde{\Delta}(T)}\right) \tag{26}$$

and for any $i = 1, \ldots, m$

$$\sum_{t=1}^{T} g_t^i(\mathbf{x}_t) \leq \mathcal{O}\left(T^{\frac{3}{4}}\widetilde{\Delta}(T)^{\frac{1}{4}}\right). \tag{27}$$

Clearly, the smaller or tighter the upper bound $\widetilde{\Delta}(T)$ is, the smaller or tighter the upper bounds for the regret and the constraint violations in (26) and (27) are. If the order of $\Delta(T)$ is known in advance, one can simply replace $\widetilde{\Delta}(T)$ with $\Delta(T)$ (with possible additional constant factor). Then, the bounds for the regret and the constraint violations are the tightest, and they indeed degenerate to their original forms in Theorem 1. Therefore, the more knowledge the agent has about $\Delta(T)$, the better the chosen stepsize and the performance guarantees are.

IV. Constrained OCO With Bandit Feedback

In this section, by exploiting some stochastic approximations and modifications, we develop a bandit version of Algorithm 1 to solve the constrained OCO with bandit feedback. The proposed algorithm only needs feedback information of the loss functions and constraint functions evaluated at two points close to the chosen actions and does not need gradients of these functions. We analyze the performance of the proposed algorithm and demonstrate that it possesses sublinear expected regret and sublinear expected constraint violations simultaneously.

A. Preliminaries

In the bandit setup, the agent only has access to the values of the loss and constraint functions at several points and does not know the gradients of these functions. As most optimization algorithms need gradients of the involved functions, a direct challenge of online algorithms with bandit feedback is how to estimate gradients based on the values of the functions at some finite number of points. To this end, given a function $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ and some small $\xi > 0$, define $\hat{\phi}(\mathbf{x}) := \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[\phi(\mathbf{x} + \xi\mathbf{v})]$ to be a smoothed version or approximation of ϕ at $\mathbf{x}$, where $\mathcal{U}(\mathcal{C})$ denotes uniform distribution over some set $\mathcal{C}$ and $\mathbb{B}$ is the unit Euclidean ball in $\mathbb{R}^n$. In [15], the following result regarding the gradient of $\hat{\phi}$ was shown.

Lemma 4: Let $\mathbb{S}$ denote the unit Euclidean sphere in $\mathbb{R}^n$. Then, we have

$$\nabla\hat{\phi}(\mathbf{x}) = \frac{n}{\xi}\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{S})}[\phi(\mathbf{x} + \xi\mathbf{u})\mathbf{u}]. \tag{28}$$

Since $\hat{\phi}$ is a smoothed approximate version of ϕ, $\nabla\hat{\phi}$ can also be regarded as an approximation of $\nabla\phi$. Thus, from Lemma 4, a reasonable estimate of $\nabla\phi(\mathbf{x})$ (and also $\nabla\hat{\phi}(\mathbf{x})$) is $\frac{n}{\xi}\phi(\mathbf{x} + \xi\mathbf{u})\mathbf{u}$, where $\mathbf{u}$ is some random vector uniformly distributed on $\mathbb{S}$. We shall make use of this estimate to develop an algorithm for constrained OCO with bandit feedback later. Prior to the formal development of the algorithm, we first present two straightforward lemmas regarding the properties of $\hat{\phi}$ in the following.

Lemma 5: If $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ is convex, then so is $\hat{\phi}$.

Lemma 6: If $\phi : \mathbb{R}^n \mapsto \mathbb{R}$ is Lipschitz continuous with constant L, i.e., $|\phi(\mathbf{x}) - \phi(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|_2 \forall \mathbf{x}, \mathbf{y}$, then so is $\hat{\phi}$.

Furthermore, we note the following two lemmas from [15] and [19], respectively, which are useful in later analysis.

Lemma 7: [15]. For any $0 < \alpha < 1$ and $\mathbf{y} \in (1-\alpha)\mathcal{X}$, we have $\mathcal{B}(\mathbf{y}, \alpha r) \subset \mathcal{X}$, where $\mathcal{B}(\mathbf{x}, a)$ denotes the Euclidean ball centered at $\mathbf{x}$ with radius a, i.e., $\mathcal{B}(\mathbf{x}, a) = \{\mathbf{z} \in \mathbb{R}^n | \|\mathbf{z} - \mathbf{x}\|_2 \leq a\}$.

Lemma 8: [19]. If ϕ_i is Lipschitz continuous with constant C_i, $i = 1, \ldots, m$, then $\phi(\mathbf{x}) = \max_{i=1,\ldots,m} \phi_i(\mathbf{x})$ is Lipschtiz continuous with constant $C = \max_{i=1,\ldots,m} C_i$.

By Assumption 4, we readily know that all loss functions f_t and constraint functions g_t^i are Lipschitz continuous with the same constant G. In other words, we have

$$|f_t(\mathbf{x}) - f_t(\mathbf{x}')| \leq G\|\mathbf{x} - \mathbf{x}'\|_2 \tag{29}$$

$$|g_t^i(\mathbf{x}) - g_t^i(\mathbf{x}')| \leq G\|\mathbf{x} - \mathbf{x}'\|_2 \tag{30}$$

for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $i = 1, \ldots, m$, $t = 1, 2, \ldots$. Define $\tilde{g}_t(\mathbf{x}) := \max_{i=1,\ldots,m} g_t^i(\mathbf{x})$, $\hat{f}_t(\mathbf{x}) := \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[f_t(\mathbf{x}+\xi\mathbf{v})]$, and $\hat{g}_t(\mathbf{x}) := \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[\tilde{g}_t(\mathbf{x}+\xi\mathbf{v})]$. According to the aforementioned lemmas, we know that $\tilde{g}_t$, $\hat{f}_t$, and $\hat{g}_t$ are all convex and Lipschitz continuous with constant G.

B. Algorithm Development

The per-slot problem can be rewritten as $\min\{f_t(\mathbf{x})|\tilde{g}_t(\mathbf{x}) \leq 0\}$. Define a modified Lagrangian associated with this problem

$$\hat{\mathfrak{L}}_t(\mathbf{x}, \lambda) := \hat{f}_t(\mathbf{x}) + \lambda\hat{g}_t(\mathbf{x}) - \frac{\eta\delta}{2}\lambda^2. \tag{31}$$

The proposed algorithm maintains and updates the primal variable (i.e., the action) $\mathbf{x}_t \in \mathbb{R}^n$ and the dual variable $\lambda_t \in \mathbb{R}$. At time t, after $\mathbf{x}_t$ is chosen, the nature reveals the values of f_t and $\tilde{g}_t$ at $\mathbf{x}_t + \xi\mathbf{u}_t$ and $\mathbf{x}_t - \xi\mathbf{u}_t$, where $\mathbf{u}_t$ is some random vector uniformly distributed over the unit sphere $\mathbb{S}$. Then, the agent performs a saddle point type of update to obtain $\mathbf{x}_{t+1}$ and λ_{t+1}. To this end, we compute the gradients of $\hat{\mathfrak{L}}_t$ as

$$\nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) = \nabla\hat{f}_t(\mathbf{x}_t) + \lambda_t\nabla\hat{g}_t(\mathbf{x}_t) \tag{32}$$

$$\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) = \hat{g}_t(\mathbf{x}_t) - \eta\delta\lambda_t. \tag{33}$$

According to Lemma 4, we have $\nabla\hat{f}_t(\mathbf{x}_t) = \frac{n}{\xi}\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{S})}[f_t(\mathbf{x}_t + \xi\mathbf{u})\mathbf{u}]$, which can be approximated as $\frac{n}{2\xi}[f_t(\mathbf{x}_t + \xi\mathbf{u}_t) - f_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u}_t$. The approximation is indeed an unbiased estimator of $\nabla\hat{f}_t(\mathbf{x}_t)$ conditioning on $\mathbf{x}_t$ as follows:

$$\frac{n}{2\xi}\mathbb{E}\{[f_t(\mathbf{x}_t + \xi\mathbf{u}_t) - f_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u}_t|\mathbf{x}_t\} = \nabla\hat{f}_t(\mathbf{x}_t). \tag{34}$$

Similarly, $\nabla\hat{g}_t(\mathbf{x}_t) = \frac{n}{\xi}\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{S})}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u})\mathbf{u}]$ can be approximated by $\frac{n}{2\xi}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) - \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u_t}$, which is also an unbiased estimator of $\nabla\hat{g}_t(\mathbf{x}_t)$, as follows:

$$\frac{n}{2\xi}\mathbb{E}\{[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) - \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u}_t|\mathbf{x}_t\} = \nabla\hat{g}_t(\mathbf{x}_t). \tag{35}$$

Furthermore, for small ξ, $\hat{g}_t(\mathbf{x}_t) = \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{v})]$ is close to $\tilde{g}_t(\mathbf{x}_t)$, which can be approximated as $\frac{1}{2}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) + \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t)]$. Combining the aforementioned approximations, we define $\mathbf{p}_t$ as an approximation of $\nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t)$ in the following

$$\begin{aligned}\mathbf{p}_t = \frac{n}{2\xi}[&f_t(\mathbf{x}_t + \xi\mathbf{u}_t) - f_t(\mathbf{x}_t - \xi\mathbf{u}_t)\\ &+ \lambda_t(\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) - \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t))]\mathbf{u}_t.\end{aligned} \tag{36}$$

We further define q_t as an approximation of $\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t)$ in the following

$$q_t = \frac{1}{2}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) + \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t)] - \eta\delta\lambda_t. \tag{37}$$

The agent performs an approximated primal descent to compute $\mathbf{x}_{t+1}$ as follows:

$$\mathbf{x}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}(\mathbf{x}_t - \eta\mathbf{p}_t) \tag{38}$$

where $\alpha \in [\frac{\xi}{r}, 1)$.[1] We initialize $\mathbf{x}_1 \in (1-\alpha)\mathcal{X}$ so that $\mathbf{x}_t \in (1-\alpha)\mathcal{X}$ for any $t = 1, 2, \ldots$. According to Lemma 7, we thus have $\mathcal{B}(\mathbf{x}_t, \alpha r) \subset \mathcal{X}$. So, $\mathbf{x}_t \pm \xi\mathbf{u}_t \in \mathcal{X}$, i.e., $\mathbf{x}_t \pm \xi\mathbf{u}_t$ are proper actions, on which f_t and $\mathbf{g}_t$ are well defined. Note that, even if the origin is not contained in the action set $\mathcal{X}$, i.e., Assumption 2 does not hold, the primal update (38) can still be used with minor modification as long as $\mathcal{X}$ contains an interior point. In such a case, for the projection set of the primal update (38), we only need to shrink the set $\mathcal{X}$ by a factor of $1-\alpha$ with respect to this interior point instead of the origin.

[1]This is a proper interval for T large enough since we will later set $\xi = o(1)$.

Algorithm 2: The Algorithm for Constrained OCO with Bandit Feedback.

1: Initialize $\mathbf{x}_1 \in (1-\alpha)\mathcal{X}$ and $\lambda_1 = 0$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Submit the action $\mathbf{x}_t$.
4: Generate $\mathbf{u}_t$ according to a uniform distribution on the unit sphere $\mathbb{S}$.
5: Query the values $f_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$ and $\tilde{g}_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$.
6: Compute $\mathbf{p}_t$ and q_t based on (36) and (37)
7: Update the primal variable, i.e., the action, according to (38) to obtain the new action $\mathbf{x}_{t+1}$.
8: Update the dual variable according to (39) to obtain the new dual variable λ_{t+1}.
9: **end for**

In addition, the agent updates the dual variable by an approximated dual ascent to obtain λ_{t+1} as

$$\lambda_{t+1} = \Pi_{\mathbb{R}^+}(\lambda_t + \eta q_t). \tag{39}$$

The proposed online algorithm for constrained OCO with bandit feedback is summarized in Algorithm 2. Later, we will set $\xi = o(1)$, i.e., ξ converges to 0 as T goes to infinity. Thus, for large T, $\mathbf{x}_t \pm \xi\mathbf{u}_t$ are two points very close to $\mathbf{x}_t$. After $\mathbf{x}_t$ is chosen, Algorithm 2 only needs the values of the loss function f_t and the maximum constraint function $\tilde{g}_t$ at $\mathbf{x}_t \pm \xi\mathbf{u}_t$ to compute the primal/dual variables at next round, i.e., $\mathbf{x}_{t+1}$ and λ_{t+1}. In other words, to operate Algorithm 2, the agent only needs bandit feedback information at two points very close to the action $\mathbf{x}_t$ and does not need gradients of the loss function and constraint functions.

C. Performance Analysis

In this subsection, we endeavor to analyze the performance of Algorithm 2 and show that it possesses sublinear regret and constraint violations as long as the drift of the benchmark sequence $\Delta(T) := \sum_{t=2}^{T}\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2$ is sublinear.

Define the function $\mathscr{H}_t : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ as

$$\begin{aligned}\mathscr{H}_t(\mathbf{x}, \lambda) := &\hat{\mathfrak{L}}_t(\mathbf{x}, \lambda) + (\mathbf{p}_t - \nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t))^{\mathsf{T}}\mathbf{x}\\ &+ \left(q_t - \frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t)\right)\lambda.\end{aligned} \tag{40}$$

For any given $\lambda \geq 0$, $\mathscr{H}_t(\cdot, \lambda)$ is convex in $\mathbf{x}$. For any given $\mathbf{x}$, $\mathscr{H}_t(\mathbf{x}, \cdot)$ is concave in λ. Furthermore, one can easily show that $\nabla_{\mathbf{x}}\mathscr{H}_t(\mathbf{x}_t, \lambda_t) = \mathbf{p}_t$ and $\frac{\partial}{\partial\lambda}\mathscr{H}_t(\mathbf{x}_t, \lambda_t) = q_t$. Thus, analogous to Lemma 2, we obtain the following lemma, the proof of which is omitted.

Lemma 9: For any $\mathbf{x} \in \mathcal{X}$, $\lambda \geq 0$, $t = 1, 2, ...$, we have

$$\begin{aligned}
&\mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}, \lambda_t) \\
&\leq \frac{1}{2\eta}\big[\|(1-\alpha)\mathbf{x} - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x} - \mathbf{x}_{t+1}\|_2^2 \\
&\quad + (\lambda - \lambda_t)^2 - (\lambda - \lambda_{t+1})^2\big] + \frac{\eta}{2}(\|\mathbf{p}_t\|_2^2 + q_t^2). \qquad (41)
\end{aligned}$$

Based on Lemma 9, we can further prove the following lemma.

Lemma 10: For any $\lambda \geq 0$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}[\mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t)] \\
&\leq \frac{4 + (1-\alpha)^2}{2\eta}R^2 + \frac{(1-\alpha)R\Delta(T)}{\eta} + \frac{\lambda^2}{2\eta} \\
&\quad + \eta T\left(n^2G^2 + D^2\right) + \eta(n^2G^2 + \eta^2\delta^2)\sum_{t=1}^{T}\lambda_t^2. \qquad (42)
\end{aligned}$$

Proof: The proof is presented in Appendix C. ■

We are now ready to show that the action sequence generated by Algorithm 2 possesses sublinear regret and constraint violations provided that the drift sequence $\Delta(\cdot)$ is sublinear.

Theorem 2: Suppose the drift sequence $\{\Delta(T')\}_{T'=1}^{\infty}$ is sublinear, i.e., $\lim_{T'\to\infty}\frac{\Delta(T')}{T'} = 0$. Assume T is large enough such that $\frac{\Delta(T)}{T} \leq \frac{1}{2(2n^2G^2+1)^2}$. Set $\xi = \frac{1}{T}$, $\alpha = \frac{1}{rT}$, $\delta = 2n^2G^2 + 1$ and $\eta = \sqrt{\frac{\Delta(T)}{T}}$. Then, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \qquad (43) \\
&\leq \left(R + n^2G^2 + D^2\right)\sqrt{T\Delta(T)} \\
&\quad + \left(\frac{5R^2}{2} + \frac{2GD}{2n^2G^2+1} + \frac{GD}{2n^2G^2+1}\left(\frac{R}{r} + 1\right)\right)\sqrt{\frac{T}{\Delta(T)}} \\
&\quad + 2G + \frac{GR}{r} \qquad (44) \\
&= \mathcal{O}\left(\sqrt{T\Delta(T)}\right) \qquad (45)
\end{aligned}$$

and for any $i = 1, \ldots, m$, the constraint violations satisfy (46)–(49) shown at the bottom of this page.

Proof: From the definition of $\mathscr{H}$ in (40), one can easily show that for any $\lambda \in \mathbb{R}$

$$\begin{aligned}
&\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda) - \hat{\mathfrak{L}}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t) \\
&= \mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t) \\
&\quad + (\mathbf{x}_t - (1-\alpha)\mathbf{x}_t^*)^{\mathsf{T}}(\nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) - \mathbf{p}_t) \\
&\quad + (\lambda - \lambda_t)\left(\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) - q_t\right). \qquad (50)
\end{aligned}$$

Summing (50) over $t = 1, \ldots, T$ and taking expectation, we obtain

$$\begin{aligned}
&\mathbb{E}\left\{\sum_{t=1}^{T}[\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda) - \hat{\mathfrak{L}}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t)]\right\} \qquad (51) \\
&= \mathbb{E}\left\{\sum_{t=1}^{T}[\mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t)]\right\} \\
&\quad + \mathbb{E}\left\{\sum_{t=1}^{T}\Big[(\mathbf{x}_t - (1-\alpha)\mathbf{x}_t^*)^{\mathsf{T}}(\nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) - \mathbb{E}[\mathbf{p}_t|\mathbf{x}_t, \lambda_t]) \right.\right. \\
&\quad \left.\left. + (\lambda - \lambda_t)\left(\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t) - \mathbb{E}[q_t|\mathbf{x}_t, \lambda_t]\right)\Big]\right\} \qquad (52)
\end{aligned}$$

where the outer expectation of the second term is with respect to $\mathbf{x}_t$ and λ_t. We note

$$\mathbb{E}[\mathbf{p}_t|\mathbf{x}_t, \lambda_t] = \nabla\hat{f}_t(\mathbf{x}_t) + \lambda_t\nabla\hat{g}_t(\mathbf{x}_t) = \nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t, \lambda_t). \qquad (53)$$

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} g_t^i(\mathbf{x}_t)\right] \qquad (46) \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\tilde{g}_t(\mathbf{x}_t)\right] \qquad (47) \\
&\leq 3G + \sqrt{2\left(\left(2n^2G^2+1\right)\sqrt{T\Delta(T)} + \sqrt{\frac{T}{\Delta(T)}}\right)} \\
&\quad \times \sqrt{FT + \left(R + n^2G^2 + D^2\right)\sqrt{T\Delta(T)} + \left(\frac{5R^2}{2} + \frac{2GD}{2n^2G^2+1} + \frac{GD}{2n^2G^2+1}\left(\frac{R}{r}+1\right)\right)\sqrt{\frac{T}{\Delta(T)}} + G\left(2 + \frac{R}{r}\right)} \qquad (48) \\
&= \mathcal{O}\left(T^{\frac{3}{4}}\Delta(T)^{\frac{1}{4}}\right). \qquad (49)
\end{aligned}$$

In addition

$$(\lambda-\lambda_t)\left(\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t)-\mathbb{E}[q_t|\mathbf{x}_t,\lambda_t]\right)$$
$$\leq|\lambda-\lambda_t|\mathbb{E}\left[\left|\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t)-q_t\right|\middle|\mathbf{x}_t,\lambda_t\right]. \tag{54}$$

We may bound λ_t as follows. According to (39) and (37), we have

$$\lambda_{t+1}\leq|\lambda_t+\eta q_t| \tag{55}$$
$$=\left|(1-\eta^2\delta)\lambda_t+\frac{\eta}{2}[\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)+\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)]\right| \tag{56}$$
$$\leq\left(1-\eta^2\delta\right)\lambda_t+\eta D. \tag{57}$$

Applying the aforementioned inequalities iteratively, we get

$$0\leq\lambda_t$$
$$\leq\left(1-\eta^2\delta\right)^{t-1}\lambda_1$$
$$+\eta D\left[1+\left(1-\eta^2\delta\right)+\cdots+\left(1-\eta^2\delta\right)^{t-2}\right]\leq\frac{D}{\eta\delta}. \tag{58}$$

Moreover, we have

$$\mathbb{E}\left[\left|\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t)-q_t\right|\middle|\mathbf{x}_t,\lambda_t\right]$$
$$=\mathbb{E}\left[\left|\hat{g}_t(\mathbf{x}_t)-\frac{1}{2}[\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)+\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)]\right|\middle|\mathbf{x}_t,\lambda_t\right] \tag{59}$$

and

$$\left|\hat{g}_t(\mathbf{x}_t)-\frac{1}{2}[\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)+\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)]\right| \tag{60}$$
$$\leq\frac{1}{2}|\hat{g}_t(\mathbf{x}_t)-\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)|+\frac{1}{2}|\hat{g}_t(\mathbf{x}_t)-\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)| \tag{61}$$
$$\leq\frac{1}{2}\mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[|\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{v})-\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)|]$$
$$+\frac{1}{2}\mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[|\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{v})-\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)|] \tag{62}$$
$$\leq 2G\xi. \tag{63}$$

Thus

$$\mathbb{E}\left[\left|\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t)-q_t\right|\middle|\mathbf{x}_t,\lambda_t\right]\leq 2G\xi. \tag{64}$$

Combining (54), (58), and (64) yields, for any $\lambda\in\mathbb{R}$

$$(\lambda-\lambda_t)\left(\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t)-\mathbb{E}[q_t|\mathbf{x}_t,\lambda_t]\right)$$
$$\leq\left(|\lambda|+\frac{D}{\eta\delta}\right)\cdot 2G\xi. \tag{65}$$

Substituting (65) and (53) into (52), we have for any $\lambda\in\mathbb{R}$

$$\mathbb{E}\left\{\sum_{t=1}^{T}[\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda)-\hat{\mathfrak{L}}_t((1-\alpha)\mathbf{x}_t^*,\lambda_t)]\right\} \tag{66}$$
$$\leq\mathbb{E}\left\{\sum_{t=1}^{T}[\mathscr{H}_t(\mathbf{x}_t,\lambda)-\mathscr{H}_t((1-\alpha)\mathbf{x}_t^*,\lambda_t)]\right\}$$
$$+2G\xi T\left(|\lambda|+\frac{D}{\eta\delta}\right). \tag{67}$$

By further using Lemma 10, we obtain that for any $\lambda\geq 0$

$$\mathbb{E}\left\{\sum_{t=1}^{T}[\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda)-\hat{\mathfrak{L}}_t((1-\alpha)\mathbf{x}_t^*,\lambda_t)]\right\}$$
$$\leq\frac{4+(1-\alpha)^2}{2\eta}R^2+\frac{(1-\alpha)R\Delta(T)}{\eta}+\frac{\lambda^2}{2\eta}$$
$$+\eta T\left(n^2G^2+D^2\right)+\eta(n^2G^2+\eta^2\delta^2)\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t^2\right]$$
$$+2G\xi T\left(\lambda+\frac{D}{\eta\delta}\right). \tag{68}$$

Substituting the definition of the modified Lagrangian $\hat{\mathfrak{L}}_t$ in (31) into (68) and rearranging terms, we get

$$\mathbb{E}\left\{\sum_{t=1}^{T}[\hat{f}_t(\mathbf{x}_t)-\hat{f}_t((1-\alpha)\mathbf{x}_t^*)]\right\}+\lambda\mathbb{E}\left[\sum_{t=1}^{T}\hat{g}_t(\mathbf{x}_t)\right]$$
$$-\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t\hat{g}_t((1-\alpha)\mathbf{x}_t^*)\right]-\frac{\eta\delta T}{2}\lambda^2+\frac{\eta\delta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t^2\right]$$
$$\leq\frac{4+(1-\alpha)^2}{2\eta}R^2+\frac{(1-\alpha)R\Delta(T)}{\eta}+\frac{\lambda^2}{2\eta}$$
$$+\eta T\left(n^2G^2+D^2\right)+\eta(n^2G^2+\eta^2\delta^2)\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t^2\right]$$
$$+2G\xi T\left(\lambda+\frac{D}{\eta\delta}\right). \tag{69}$$

According to our choice of δ and η, it can be easily verified that $\eta(n^2G^2+\eta^2\delta^2)-\frac{\eta\delta}{2}\leq 0$. Hence, for any $\lambda\geq 0$

$$\mathbb{E}\left\{\sum_{t=1}^{T}[\hat{f}_t(\mathbf{x}_t)-\hat{f}_t((1-\alpha)\mathbf{x}_t^*)]\right\}+\lambda\mathbb{E}\left[\sum_{t=1}^{T}\hat{g}_t(\mathbf{x}_t)\right]$$
$$-\mathbb{E}\left[\sum_{t=1}^{T}\lambda_t\hat{g}_t((1-\alpha)\mathbf{x}_t^*)\right]-\frac{\eta\delta T}{2}\lambda^2$$
$$\leq\frac{4+(1-\alpha)^2}{2\eta}R^2+\frac{(1-\alpha)R\Delta(T)}{\eta}+\frac{\lambda^2}{2\eta}$$
$$+\eta T\left(n^2G^2+D^2\right)+2G\xi T\left(\lambda+\frac{D}{\eta\delta}\right). \tag{70}$$

To relate the left-hand side (L.H.S.) of (70) with the regret $\texttt{Reg}(T)$ and the constraint violations $\texttt{Vio}^i(T)$, we endeavor to replace $\hat{f}_t$ and $\hat{g}_t$ on the L.H.S. of (70) with f_t and $\tilde{g}_t$, respectively. To this end, we have

$$|\hat{f}_t(\mathbf{x}_t)-f_t(\mathbf{x}_t)|\leq\mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[|f_t(\mathbf{x}_t+\xi\mathbf{v})-f_t(\mathbf{x}_t)|]\leq\xi G. \tag{71}$$

So

$$\hat{f}_t(\mathbf{x}_t) \geq f_t(\mathbf{x}_t) - \xi G. \tag{72}$$

In addition

$$\begin{aligned}
&|\hat{f}_t((1-\alpha)\mathbf{x}_t^*) - f_t(\mathbf{x}_t^*)| && (73)\\
&\leq |\hat{f}_t((1-\alpha)\mathbf{x}_t^*) - f_t((1-\alpha)\mathbf{x}_t^*)| \\
&\quad + |f_t((1-\alpha)\mathbf{x}_t^*) - f_t(\mathbf{x}_t^*)| && (74)\\
&\leq |\mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[f_t((1-\alpha)\mathbf{x}_t^* + \xi\mathbf{v}) - f_t((1-\alpha)\mathbf{x}_t^*)]| \\
&\quad + G\alpha\|\mathbf{x}_t^*\|_2 && (75)\\
&\leq G\xi + G\alpha R. && (76)
\end{aligned}$$

Thus

$$\hat{f}_t((1-\alpha)\mathbf{x}_t^*) \leq f_t(\mathbf{x}_t^*) + G\alpha R + G\xi. \tag{77}$$

Similarly, we can show

$$\hat{g}_t(\mathbf{x}_t) \geq \tilde{g}_t(\mathbf{x}_t) - G\xi \tag{78}$$

and

$$\hat{g}_t((1-\alpha)\mathbf{x}_t^*) \leq \tilde{g}_t(\mathbf{x}_t^*) + G\alpha R + G\xi. \tag{79}$$

Multiplying both sides of (79) with $\lambda_t \geq 0$, summing over $t = 1, \ldots, T$ and noting that $\tilde{g}_t(\mathbf{x}_t^*) \leq 0$, we get

$$\begin{aligned}
&\sum_{t=1}^{T} \lambda_t \hat{g}_t((1-\alpha)\mathbf{x}_t^*) \\
&\leq \sum_{t=1}^{T} \lambda_t \tilde{g}_t(\mathbf{x}_t^*) + (G\alpha R + G\xi)\sum_{t=1}^{T}\lambda_t \\
&\leq (G\alpha R + G\xi)\sum_{t=1}^{T}\lambda_t. && (80)
\end{aligned}$$

Substituting (72), (77), (78), and (80) into (70), noting that $\mathbb{E}[\sum_{t=1}^{T}\lambda_t] \leq \frac{DT}{\eta\delta}$ due to (58), and rearranging terms, we have for any $\lambda \geq 0$

$$\begin{aligned}
&\mathbb{E}\left\{\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]\right\} - \left(\frac{\eta\delta T}{2} + \frac{1}{2\eta}\right)\lambda^2 \\
&+ \left(\mathbb{E}\left[\sum_{t=1}^{T}\tilde{g}_t(\mathbf{x}_t)\right] - 3G\xi T\right)\lambda \\
&\leq \frac{4+(1-\alpha)^2}{2\eta}R^2 + \frac{(1-\alpha)R\Delta(T)}{\eta} + \eta T\left(n^2G^2 + D^2\right) \\
&+ \frac{2G\xi TD}{\eta\delta} + (2\xi G + G\alpha R)T + (G\alpha R + G\xi)\frac{DT}{\eta\delta}. && (81)
\end{aligned}$$

Setting $\lambda = 0$ in (81) and substituting the $\xi, \alpha, \eta,$ and δ with the specified values in the statement of the theorem, we obtain the desired bound on regret in (44) and (45). In addition, setting $\lambda = \frac{(\mathbb{E}[\sum_{t=1}^{T}\tilde{g}_t(\mathbf{x}_t)] - 3G\xi T)^+}{\eta\delta T + \frac{1}{\eta}}$ (calculated by maximizing the L.H.S. of (81) with respect to $\lambda \geq 0$) in (81) and noting that $\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)] \geq -FT$ due to Assumption 6, with the specified values for $\xi, \alpha, \eta,$ and δ, we can get the desired bound on constraint violations in (48) and (49). ■

Remark 3: Theorem 2 asserts that Algorithm 2 achieves the same performance scaling (in terms of regret and constraint violations) for constrained OCO with bandit feedback as Algorithm 1 does for gradient feedback. This implies that bandit feedback about loss/constraint functions does not hurt much for constrained OCO with time-varying constraints.

Analogous to Theorem 1, the choice of the stepsize η in Theorem 2 also relies on the drift $\Delta(T)$, which may not be known to the agent. Nevertheless, similar to Corollary 1, if a sublinear upper bound on $\Delta(T)$, denoted as $\widetilde{\Delta}(T)$, is available to the agent, we can still guarantee the sublinearity of the regret and constraint violations of Algorithm 2 by changing the stepsize to be $\eta = \sqrt{\frac{\widetilde{\Delta}(T)}{T}}$. Formally, this is summarized in the following corollary.

Corollary 2: Suppose the drift sequence $\Delta(T)$ is sublinear. Furthermore, presume that, before the start of the OCO, the agent knows a sublinear positive sequence $\widetilde{\Delta}(T)$ such that $\Delta(T) \leq \widetilde{\Delta}(T)$ for T large enough. Set $\xi = \frac{1}{T}$, $\alpha = \frac{1}{rT}$, $\delta = 2n^2G^2 + 1$, and $\eta = \sqrt{\frac{\widetilde{\Delta}(T)}{T}}$. Then, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \leq \mathcal{O}\left(\sqrt{T\widetilde{\Delta}(T)}\right) \tag{82}$$

and for any $i = 1, \ldots, m$

$$\mathbb{E}\left[\sum_{t=1}^{T} g_t^i(\mathbf{x}_t)\right] \leq \mathcal{O}\left(T^{\frac{3}{4}}\widetilde{\Delta}(T)^{\frac{1}{4}}\right). \tag{83}$$

The remarks of Corollary 1 also applies to Corollary 2. If the agent has more knowledge of the drift $\Delta(T)$, a smaller or tighter upper bound $\widetilde{\Delta}(T)$ is available, which in turn leads to better performance guarantees in (82) and (83). If the agent knows little about $\Delta(T)$ besides sublinearity, she can choose a conservative but still sublinear upper bound $\widetilde{\Delta}(T)$, e.g., T^{ζ} with $\zeta < 1$ close to 1. In such a case, sublinearity of the regret and the constraint violations can still be guaranteed.

V. Numerical Experiments

In this section, numerical experiments are conducted to corroborate the effectiveness of the proposed algorithms for the constrained OCO with gradient feedback (Algorithm 1) or bandit feedback (Algorithm 2). Specifically, we study two numerical examples: OQP and OLR. We empirically observe that, in either example, for both gradient feedback and bandit feedback, as time goes to infinity, the time average regrets converge to zero and the time average constraint violations become negative, confirming the theoretical guarantees in Theorems 1 and 2.

A. Online Quadratic Programming (OQP)

In this subsection, we study a numerical example of the OQP. The optimization problem of the OQP at time t is

$$\begin{aligned}
&\underset{\mathbf{x}\in\mathcal{X}}{\text{Minimize}} \ \ \mathbf{x}^{\mathsf{T}}\mathbf{A}_t\mathbf{x} + \mathbf{b}_t^{\mathsf{T}}\mathbf{x} \\
&\text{s.t.} \ \ \mathbf{C}_t\mathbf{x} \preceq \mathbf{d}_t
\end{aligned} \tag{84}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable; $\mathbf{A}_t \in \mathbb{R}^{n\times n}$ is some positive semidefinite matrix; $\mathbf{b}_t \in \mathbb{R}^n$, $\mathbf{C}_t \in \mathbb{R}^{m\times n}$, and $\mathbf{d}_t \in \mathbb{R}^m$; and $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_2 \leq R\}$ is the action set with R some positive number. The problem (84) is in the form of (1) with $f_t(\mathbf{x}) = \mathbf{x}^\mathsf{T}\mathbf{A}_t\mathbf{x} + \mathbf{b}_t^\mathsf{T}\mathbf{x}$ and $\mathbf{g}_t(\mathbf{x}_t) = \mathbf{C}_t\mathbf{x} - \mathbf{d}_t$. As stated in Section II, at time t, when making decision $\mathbf{x}_t$, the agent is unaware of the problem data, i.e., $\mathbf{A}_t$, $\mathbf{b}_t$, $\mathbf{C}_t$, and $\mathbf{d}_t$. With the gradient feedback, after $\mathbf{x}_t$ is submitted, all problem data $\mathbf{A}_t$, $\mathbf{b}_t$, $\mathbf{C}_t$, and $\mathbf{d}_t$ will be revealed to the agent. In contrast, with the bandit feedback, after $\mathbf{x}_t$ is submitted, only $f_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$ and $\tilde{g}_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$ are announced to the agent. Such an OQP formulation has broad applications in many signal processing and control problems. For instance, in dynamic resource allocation, the OQP is to minimize some quadratic costs while satisfying some linear resource constraints. The cost functions and resource constraints are time varying and unknown *a priori* due to factors such as the uncertainty of renewable energy supply and electricity market prices in smart grids. As another application, the OQP can also correspond to dynamic linear regression with linear constraints where the fitting data or observations are time varying and unknown *a priori* (possibly due to the delay of data/observations).

The time-varying problem data are generated as follows. With $\mathbf{A}_t, \mathbf{b}_t, \mathbf{C}_t,$ and $\mathbf{d}_t$ in hands, our goal is to generate $\mathbf{A}_{t+1}, \mathbf{b}_{t+1}, \mathbf{C}_{t+1},$ and $\mathbf{d}_{t+1}$. To obtain $\mathbf{A}_{t+1}$, we first generate $\tilde{\mathbf{A}}_t := \mathbf{A}_t + \mathbf{W}_t$, where $\mathbf{W}_t$ is some symmetric disturbance matrix. For $1 \leq i \leq j \leq n$, $W_{t,ij}$ is generated according to uniform distribution on $[-\frac{1}{2t}, \frac{1}{2t}]$ independently. For $i > j$, $W_{t,ij} = W_{t,ji}$. Afterwards, $\mathbf{A}_{t+1}$ is the projection of $\tilde{\mathbf{A}}_t$ onto the positive semidefinite cone, i.e., the set of positive semidefinite matrices. This projection involves eigendecomposition of $\tilde{\mathbf{A}}_t$. To obtain $\mathbf{b}_{t+1}$ and $\mathbf{C}_{t+1}$, we similarly add disturbance vectors/matrices onto $\mathbf{b}_t$ and $\mathbf{C}_t$, respectively, and each entry of the disturbance vectors/matrices is uniformly distributed on $[-\frac{1}{2t}, \frac{1}{2t}]$. To update the sequence of $\mathbf{d}_t$, we introduce an auxiliary sequence $\mathbf{x}_t^\# \in \mathbb{R}^n$, where $\mathbf{x}_{t+1}^\#$ is the sum of $\mathbf{x}_t^\#$ and a disturbance vector with each entry uniformly distributed on $[-\frac{1}{2t}, \frac{1}{2t}]$. Then, $\mathbf{d}_t$ is computed as $\mathbf{d}_t = \mathbf{C}_t\mathbf{x}_t^\# + \mathbf{r}_t$, where each entry of $\mathbf{r}_t$ is uniformly distributed on $[0, 1]$. The problem data are initialized as follows. $\mathbf{A}_1 = \mathbf{I}$. Each entry of $\mathbf{b}_1$ is uniformly distributed on $[-0.5, 0.5]$. Each entry of $\mathbf{C}_1$ is uniformly distributed on $[0, 1]$. Each entry of $\mathbf{x}_1^\#$ is uniformly distributed on $[-1, 0]$. The parameters are set according to the specifications in Theorems 1 and 2 as follows: $m = 3, n = 10, T = 1000,\ R = 5,\ \eta = \frac{1}{\sqrt{T}},\ \delta = 10,\ \xi = \frac{1}{T},$ $r = \frac{R}{2}$, and $\alpha = \frac{1}{rT}$. In particular, since the data evolution rate is $\frac{1}{t}$, so is the order of the gap between adjacent benchmark optima, i.e., $\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 = \Theta(\frac{1}{t})$. Thus, according to the definition of drift, we know $\Delta(T) = \sum_{t=2}^T \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 = \Theta(\log T)$. So, the stepsize prescribed by Theorems 1 and 2 is $\sqrt{\frac{\Delta(T)}{T}} = \Theta(\sqrt{\frac{\log T}{T}})$, which will lead to sublinear regret bound of order $\mathcal{O}(\sqrt{T\Delta(T)}) = \mathcal{O}(\sqrt{T\log T})$ and sublinear constraint violation bound of order $\mathcal{O}(T^{\frac{3}{4}}\Delta(T)^{\frac{1}{4}}) = \mathcal{O}(T^{\frac{3}{4}}(\log T)^{\frac{1}{4}})$. We note that the logarithmic term does not affect the stepsize much due to the presence of T in the denominator. So, we can omit the term $\log T$ and simply set the stepsize as $\eta = \frac{1}{\sqrt{T}}$, which can still ensure sublinear regrets and constraint violations (adding logarithmic terms into the regret and constraint violation bounds will not affect their sublinearity).

We apply the proposed algorithms, i.e., Algorithms 1 and 2, to the OQP and the time average regrets $\frac{\texttt{Reg}(t)}{t}$ and time average constraint violations $\frac{\texttt{Vio}^i(t)}{t}$ are shown in Fig. 1(a) and (b), respectively. The scenarios of both gradient feedback and bandit feedback are considered. In both scenarios, we observe that, as time progresses, the time average regrets converge to zero and the time average constraint violations become negative, in accordance with the analytical results in Theorems 1 and 2. To investigate the impact of the evolution rate of the time-varying problem data, we alter the distribution of all the disturbances to be uniform distribution over the interval $[-\frac{1}{2\sqrt{t}}, \frac{1}{2\sqrt{t}}]$, which increases the evolution speed of the problem data. According to Theorems 1 and 2, the stepsize $\eta = \sqrt{\frac{\Delta(T)}{T}}$ is changed to be $0.2T^{-\frac{1}{4}}$, since in this scenario, $\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 = \Theta(\frac{1}{\sqrt{t}})$ and $\Delta(T) = \sum_{t=2}^T \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 = \Theta(\sqrt{T})$. This will lead to sublinear regret bound of order $\mathcal{O}(T^{\frac{3}{4}})$ and sublinear constraint violation bound of order $\mathcal{O}(T^{\frac{7}{8}})$ theoretically. The corresponding time average regrets and time average constraint violations are reported in Fig. 1(c) and (d), respectively. In Fig. 1(c), we observe that the time average regrets are larger than that of the $\frac{1}{t}$ evolution rate [see Fig. 1(a)], especially for the scenario of the bandit feedback. However, the time average regrets can still converge to zero, as guaranteed theoretically. Comparing Fig. 1(d) with 1(b), we remark that the constraint violations do not change much and are still negative as time approaches infinity.

The numerical results for the bandit feedback shown in Fig. 1 are for one particular realization of the random query points at which the values of the loss/constraint functions are revealed. In Fig. 2, we further study the *average* performance of Algorithm 2 for the bandit feedback by 1000 independent Monte-Carlo trials. The data evolution rate is set to be $\frac{1}{\sqrt{t}}$ and the corresponding performance of Algorithm 1 for the gradient feedback is also shown for comparison. We observe that, similar to the one-shot realization results in Fig. 1, the averaged (over multiple trials) time-average regrets of the bandit feedback still converge to zero and the averaged time-average constraint violations of bandit feedback are still asymptotically nonpositive. In addition, the average performance of the bandit feedback is very close to that of the gradient feedback, verifying that the bandit feedback does not affect the performance of constrained OCO much (c.f., Remark 3).

B. Online Logistic Regression (OLR)

In this subsection, we investigate a numerical example of OLR. The optimization problem of OLR at time t is

$$\begin{aligned} \underset{\mathbf{x}\in\mathcal{X}}{\text{Minimize}} \quad & \sum_{i=1}^{k} \log\left(1 + \exp\left(-l_{i,t}\mathbf{u}_{i,t}^\mathsf{T}\mathbf{x}\right)\right) \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \leq a_t \end{aligned} \tag{85}$$

where $\mathbf{u}_{i,t} \in \mathbb{R}^n$ is the ith training point at time t and $l_{i,t} \in \{-1, 1\}$ is the corresponding label. a_t is a threshold on the

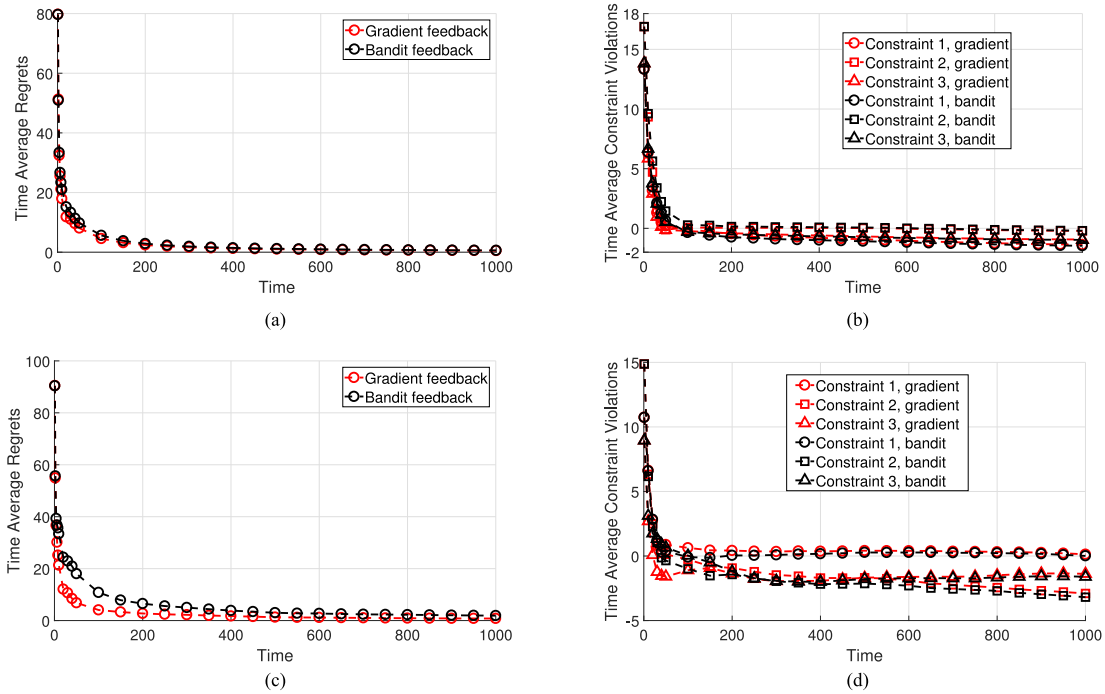


Fig. 1. Regrets and constraint violations for OQP with different problem data evolution rates. Scenarios of both gradient feedback and bandit feedback are considered. (a) Regrets of OQP with problem data evolution rate equal to $\frac{1}{t}$. (b) Constraint violations of OQP with problem data evolution rate equal to $\frac{1}{t}$. (c) Regrets of OQP with problem data evolution rate equal to $\frac{1}{\sqrt{t}}$. (d) Constraint violations of OQP with problem data evolution rate equal to $\frac{1}{\sqrt{t}}$.

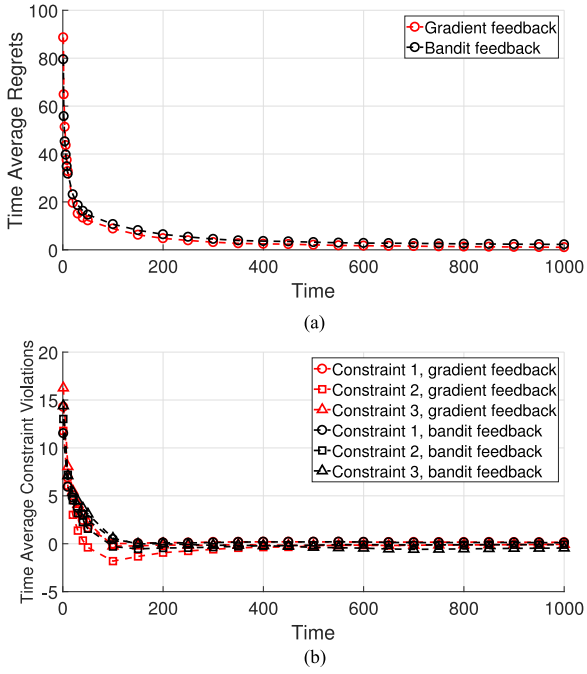


Fig. 2. Regrets and constraint violations for OQP with data evolution rate equal to $\frac{1}{\sqrt{t}}$, in which the results for the bandit feedback are averaged over 1000 Monte-Carlo trials. (a) Regrets of OQP. (b) Constraint violations of OQP.

l_1 norm of the weight vector $\mathbf{x}$ to enforce sparsity. $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_\infty \leq M\}$ is the action set and M is some positive number. Such an OLR problem (85) is in the form of (1) with $f_t(\mathbf{x}) = \sum_{i=1}^k \log(1 + \exp(-l_{i,t}\mathbf{u}_{i,t}^\mathsf{T}\mathbf{x}))$ and $g_t(\mathbf{x}) = \|\mathbf{x}\|_1 - a_t$. When deciding $\mathbf{x}_t$, the agent does not know the problem data $\{\mathbf{u}_{i,t}, l_{i,t}\}_{i=1}^k$ and a_t, possibly due to the delay of the training data.

The problem data $\mathbf{u}_{i,t}, l_{i,t}$, and a_t are generated according to the following procedure. $\mathbf{u}_{i,t+1} = \mathbf{u}_{i,t} + \boldsymbol{\beta}_{i,t}$, where each entry of $\boldsymbol{\beta}_{i,t} \in \mathbb{R}^n$ is uniformly distributed over $[-\frac{1}{2t}, \frac{1}{2t}]$. $a_{t+1} = (a_t + \gamma_t)^+$, where $x^+ := \max\{0, x\}$ and γ_t is uniformly distributed over $[-\frac{1}{2t}, \frac{1}{2t}]$. Additionally, we generate an auxiliary true weight vector sequence $\mathbf{x}_t^\# \in \mathbb{R}^n$ for the OLR. The true weight vector sequence is updated as follows. Given $\mathbf{x}_t^\#$, we compute $\check{\mathbf{x}}_t = \mathbf{x}_t^\# + \boldsymbol{\tau}_t$, where each entry of $\boldsymbol{\tau}_t \in \mathbb{R}^n$ is uniformly distributed over the interval $[-\frac{1}{2t}, \frac{1}{2t}]$. Compute $\tilde{\mathbf{x}}_t = \Pi_{\mathcal{X}}(\check{\mathbf{x}}_t)$. Then, if $\|\tilde{\mathbf{x}}_t\|_1 \leq a_{t+1}$, we set $\mathbf{x}_{t+1}^\# = \tilde{\mathbf{x}}_t$. Otherwise, set $\mathbf{x}_{t+1}^\# = \frac{a_{t+1}}{\|\tilde{\mathbf{x}}_t\|_1}\tilde{\mathbf{x}}_t$. The parameters of the problem are set as follows: $n = 5, k = 20, T = 10000, \eta = 0.2\frac{1}{\sqrt{T}}, \delta = 10, \xi = \frac{1}{T}, r = M = 30,$ and $\alpha = \frac{1}{rT}$.

We apply the proposed algorithms, i.e., Algorithms 1 and 2, to the OLR and the time average regrets $\frac{\mathtt{Reg}(t)}{t}$ and time average constraint violations $\frac{\mathtt{Vio}^i(t)}{t}$ are shown in Fig. 3(a) and (b), respectively. We remark that, for both gradient feedback and bandit feedback, as time goes to infinity, the time average regrets converge to zero and the time average constraint violations become negative. This confirms the theoretical guarantees in Theorems 1 and 2 again. Furthermore, we investigate the tracking errors with respect to the true weight vectors $\mathbf{x}_t^\#$. In Fig. 3(c), we plot the tracking errors of the gradient feedback ($\|\mathbf{x}_t - \mathbf{x}_t^\#\|_2$, where $\mathbf{x}_t$ is generated by Algorithm 1 with gradi-

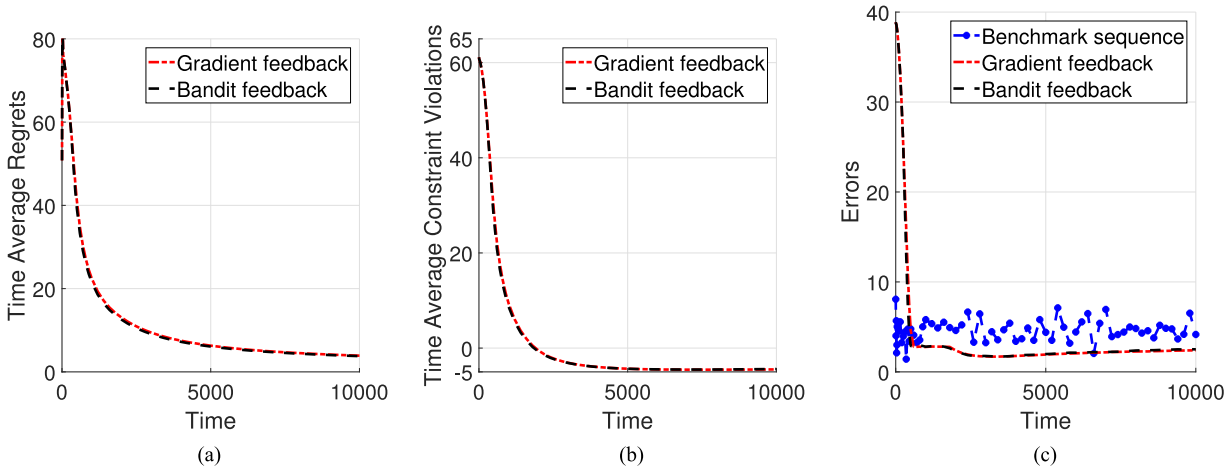


Fig. 3. Regrets, constraint violations, and errors with respect to the true weight vectors for OLR. (a) Regrets of OLR. (b) Constraint violations of OLR. (c) Errors with respect to the true weight vectors $\mathbf{x}_t^{\#}$.

ent feedback), bandit feedback ($\|\mathbf{x}_t - \mathbf{x}_t^{\#}\|_2$, where $\mathbf{x}_t$ is generated by Algorithm 2 with bandit feedback), and the benchmark sequence ($\|\mathbf{x}_t^* - \mathbf{x}_t^{\#}\|_2$, where $\mathbf{x}_t^*$ is the optimal point of (85), i.e., the benchmark or the posteriori optima). From Fig. 3(c), we observe that, for both gradient feedback and bandit feedback, the proposed algorithms can track the true weight vectors well after about 500 time slots. We remark that, once stable, the tracking errors of the proposed algorithms are less than that of the benchmark sequence $\mathbf{x}_t^*$. The reason is that the proposed online algorithms take information (training data) from previous time into account, while the benchmark $\mathbf{x}_t^*$ is computed solely based on the data of the current time instant.

VI. Conclusion

In this paper, we study constrained OCO problems with time-varying constraints. For the gradient feedback, we propose a computationally efficient online algorithm (Algorithm 1), which only involves direct closed-form computations at each time instant. We establish sublinear regret and constraint violations of Algorithm 1 under the assumption that the drift of the benchmark sequence is sublinear, i.e., the underlying dynamic optimization problem does not vary too fast. Moreover, we investigate a bandit version of the constrained OCO problem and propose an online algorithm (Algorithm 2) for the bandit feedback. Analogous sublinear results for the expected regrets and the expected constraint violations of Algorithm 2 are demonstrated. Finally, numerical examples of OQP and OLR are presented to validate the effectiveness of the proposed algorithms.

Appendix A
Proof of Lemma 2

Given any $\boldsymbol{\lambda}' \succeq \mathbf{0}$, $\mathfrak{L}_t(\cdot, \boldsymbol{\lambda}')$ is convex in $\mathbf{x}$. Hence

$$\mathfrak{L}_t(\mathbf{x}, \boldsymbol{\lambda}_t) \geq \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) + \nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)^{\mathsf{T}}(\mathbf{x} - \mathbf{x}_t). \tag{86}$$

Similarly, given any $\mathbf{x}'$, $\mathfrak{L}_t(\mathbf{x}', \cdot)$ is concave in $\boldsymbol{\lambda}$. Thus

$$\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) \leq \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) + \nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)^{\mathsf{T}}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_t). \tag{87}$$

Combining (86) and (87) yields

$$\begin{aligned}&\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \mathfrak{L}_t(\mathbf{x}, \boldsymbol{\lambda}_t)\\ &\leq \nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}) + \nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)^{\mathsf{T}}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_t).\end{aligned} \tag{88}$$

According to the update of the primal variable in (5), Lemma 1, and $\mathbf{x} \in \mathcal{X}$, we obtain

$$\|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2 \leq \|\mathbf{x} - \mathbf{x}_t + \eta \nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 \tag{89}$$

which can be rewritten as

$$\begin{aligned}&2\eta(\mathbf{x}_t - \mathbf{x})^{\mathsf{T}} \nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\\ &\leq \|\mathbf{x} - \mathbf{x}_t\|_2^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2 + \eta^2 \|\nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2.\end{aligned} \tag{90}$$

Similarly, based on the update of the dual variable in (7), Lemma 1, and $\boldsymbol{\lambda} \succeq \mathbf{0}$, we obtain

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t+1}\|_2^2 \leq \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t - \eta \nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 \tag{91}$$

which can be rewritten as

$$\begin{aligned}&2\eta(\boldsymbol{\lambda} - \boldsymbol{\lambda}_t)^{\mathsf{T}} \nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\\ &\leq \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t\|_2^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t+1}\|_2^2 + \eta^2 \|\nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2.\end{aligned} \tag{92}$$

Substituting (90) and (92) into (88) yields the desired result in (9).

Appendix B
Proof of Lemma 3

According to Lemma 2 and noting that $\mathbf{x}_t^* \in \mathcal{X}$, we have

$$\begin{aligned}&\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \mathfrak{L}_t(\mathbf{x}_t^*, \boldsymbol{\lambda}_t)\\ &\leq \frac{1}{2\eta}\left(\|\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2 + \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_t\|_2^2 - \|\boldsymbol{\lambda}\right.\\ &\left. - \boldsymbol{\lambda}_{t+1}\|_2^2\right) + \frac{\eta}{2}\left(\|\nabla_{\mathbf{x}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 + \|\nabla_{\boldsymbol{\lambda}} \mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2\right).\end{aligned} \tag{93}$$

Summing (93) over the time period of length T, at the right-hand side, we can make use of the telescoping sum $\sum_{t=1}^{T}(\|\boldsymbol{\lambda} -$

$\boldsymbol{\lambda}_t\|_2^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t+1}\|_2^2) = \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_1\|_2^2 - \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{T+1}\|_2^2 \leq \|\boldsymbol{\lambda}\|_2^2$ due to the initial condition $\boldsymbol{\lambda}_1 = \mathbf{0}$ and the nonnegativity of $\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{T+1}\|_2^2$. Thus, we have

$$\begin{aligned}
&\sum_{t=1}^{T}[\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \mathfrak{L}_t(\mathbf{x}_t^*, \boldsymbol{\lambda}_t)] \\
&\leq \frac{1}{2\eta}\sum_{t=1}^{T}\left(\|\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2\right) + \frac{1}{2\eta}\|\boldsymbol{\lambda}\|_2^2 \\
&\quad + \frac{\eta}{2}\sum_{t=1}^{T}\left(\|\nabla_{\mathbf{x}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 + \|\nabla_{\boldsymbol{\lambda}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2\right). \qquad (94)
\end{aligned}$$

For the first term of (94), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\|\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2\right) \\
&= \|\mathbf{x}_1^* - \mathbf{x}_1\|_2^2 - \|\mathbf{x}_T^* - \mathbf{x}_{T+1}\|_2^2 \\
&\quad + \sum_{t=2}^{T}\left(\|\mathbf{x}_t - \mathbf{x}_t^*\|_2^2 - \|\mathbf{x}_t - \mathbf{x}_{t-1}^*\|_2^2\right) \qquad (95) \\
&\leq \|\mathbf{x}_1^* - \mathbf{x}_1\|_2^2 + \sum_{t=2}^{T} 2\mathbf{x}_t^{\mathsf{T}}(\mathbf{x}_{t-1}^* - \mathbf{x}_t^*) \\
&\quad + \sum_{t=2}^{T}\left(\|\mathbf{x}_t^*\|_2^2 - \|\mathbf{x}_{t-1}^*\|_2^2\right) \qquad (96) \\
&\leq \|\mathbf{x}_1^* - \mathbf{x}_1\|_2^2 + \|\mathbf{x}_T^*\|_2^2 + 2\sum_{t=2}^{T}\|\mathbf{x}_t\|_2\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 \qquad (97) \\
&\leq 5R^2 + 2R\Delta(T) \qquad (98)
\end{aligned}$$

where we make use of Assumption 2 in the last step. As for the last term regarding gradients of the modified Lagrangian, we have

$$\begin{aligned}
&\|\nabla_{\mathbf{x}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 \\
&= \left\|\nabla f_t(\mathbf{x}_t) + \sum_{i=1}^{m}\lambda_t^i \nabla g_t^i(\mathbf{x}_t)\right\|_2^2 \qquad (99) \\
&\leq \left(\|\nabla f_t(\mathbf{x}_t)\|_2 + \sum_{i=1}^{m}\lambda_t^i\|\nabla g_t^i(\mathbf{x}_t)\|_2\right)^2 \qquad (100) \\
&\leq (1+m)\left[\|\nabla f_t(\mathbf{x}_t)\|_2^2 + \sum_{i=1}^{m}\left(\lambda_t^i\right)^2\|\nabla g_t^i(\mathbf{x}_t)\|_2^2\right] \qquad (101) \\
&\leq (1+m)G^2(1 + \|\boldsymbol{\lambda}_t\|_2^2) \qquad (102)
\end{aligned}$$

where we use Assumption 4 in the last step. Additionally

$$\begin{aligned}
&\|\nabla_{\boldsymbol{\lambda}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\|_2^2 \\
&= \|\mathbf{g}_t(\mathbf{x}_t) - \delta\eta\boldsymbol{\lambda}_t\|_2^2 \qquad (103) \\
&\leq 2\left(\|\mathbf{g}_t(\mathbf{x}_t)\|_2^2 + \delta^2\eta^2\|\boldsymbol{\lambda}\|_2^2\right) \qquad (104) \\
&\leq 2\left(D^2 + \delta^2\eta^2\|\boldsymbol{\lambda}_t\|_2^2\right) \qquad (105)
\end{aligned}$$

where we use Assumption 5 in the last step. Substituting (98), (102), and (105) into (94) and rearranging terms yield the desired result in (10).

APPENDIX C
PROOF OF LEMMA 10

According to Lemma 9, for any $\lambda \geq 0$, we have

$$\begin{aligned}
&\mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t) \\
&\leq \frac{1}{2\eta}\left[\|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2\right. \\
&\quad \left. + (\lambda - \lambda_t)^2 - (\lambda - \lambda_{t+1})^2\right] + \frac{\eta}{2}(\|\mathbf{p}_t\|_2^2 + q_t^2). \qquad (106)
\end{aligned}$$

Summing (106) over $t = 1, \ldots, T$ and making use of the telescoping sum $\sum_{t=1}^{T}[(\lambda - \lambda_t)^2 - (\lambda - \lambda_{t+1})^2] = (\lambda - \lambda_1)^2 - (\lambda - \lambda_{T+1})^2 \leq \lambda^2$ (since $\lambda_1 = 0$), we obtain

$$\begin{aligned}
&\sum_{t=1}^{T}[\mathscr{H}_t(\mathbf{x}_t, \lambda) - \mathscr{H}_t((1-\alpha)\mathbf{x}_t^*, \lambda_t)] \\
&\leq \frac{1}{2\eta}\sum_{t=1}^{T}\left[\|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2\right] \\
&\quad + \frac{\lambda^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}(\|\mathbf{p}_t\|_2^2 + q_t^2). \qquad (107)
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&\sum_{t=2}^{T}\left[\|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x}_{t-1}^* - \mathbf{x}_t\|_2^2\right] \qquad (108) \\
&= (1-\alpha)^2\left[\|\mathbf{x}_T^*\|_2^2 - \|\mathbf{x}_1^*\|_2^2\right] + 2(1-\alpha)\sum_{t=2}^{T}\mathbf{x}_t^{\mathsf{T}}(\mathbf{x}_{t-1}^* - \mathbf{x}_t^*) \qquad (109) \\
&\leq (1-\alpha)^2R^2 + 2(1-\alpha)\sum_{t=2}^{T}\|\mathbf{x}_t\|_2\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|_2 \qquad (110) \\
&\leq (1-\alpha)^2R^2 + 2(1-\alpha)R\Delta(T). \qquad (111)
\end{aligned}$$

Hence, for the first term of (107), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\left[\|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_{t+1}\|_2^2\right] \qquad (112) \\
&= \|(1-\alpha)\mathbf{x}_1^* - \mathbf{x}_1\|_2^2 - \|(1-\alpha)\mathbf{x}_T^* - \mathbf{x}_{T+1}\|_2^2 \\
&\quad + \sum_{t=2}^{T}\left[\|(1-\alpha)\mathbf{x}_t^* - \mathbf{x}_t\|_2^2 - \|(1-\alpha)\mathbf{x}_{t-1}^* - \mathbf{x}_t\|_2^2\right] \qquad (113) \\
&\leq 4R^2 + (1-\alpha)^2R^2 + 2(1-\alpha)R\Delta(T). \qquad (114)
\end{aligned}$$

According to (36)

$$\begin{aligned}
&\|\mathbf{p}_t\|_2 \\
&= \frac{n}{2\xi}|f_t(\mathbf{x}_t + \xi\mathbf{u}_t) - f_t(\mathbf{x}_t - \xi\mathbf{u}_t) \\
&\quad + \lambda_t(\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) - \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t))| \qquad (115)
\end{aligned}$$

$$\leq \frac{n}{2\xi}|f_t(\mathbf{x}_t+\xi\mathbf{u}_t)-f_t(\mathbf{x}_t-\xi\mathbf{u}_t)| + \frac{n\lambda_t}{2\xi}|\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)-\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)| \quad (116)$$

$$\leq \frac{nG}{2\xi}\cdot 2\xi + \frac{nG\lambda_t}{2\xi}\cdot 2\xi \quad (117)$$

$$= nG(1+\lambda_t) \quad (118)$$

where we make use of Assumption 4 to get (117). So

$$\|\mathbf{p}_t\|_2^2 \leq 2n^2G^2(1+\lambda_t^2). \quad (119)$$

In addition, according to (37), we know

$$|q_t| \leq \frac{1}{2}|\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)+\tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)| + \eta\delta\lambda_t \leq D+\eta\delta\lambda_t \quad (120)$$

where we have used Assumption 5. Thus

$$q_t^2 \leq 2\left(D^2+\eta^2\delta^2\lambda_t^2\right). \quad (121)$$

Substituting (114), (119), and (121) into (107), we get the desired result in (42).

REFERENCES

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

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

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

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

[5] 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.

[6] 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.*, New York, NY, USA, 2013, pp. 53–66.

[7] 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.

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

[9] 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," *Perform. Eval.*, vol. 70, no. 10, pp. 770–791, 2013.

[10] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–935.

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

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

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

[14] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 7195–7201.

[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. 23rd Annu. 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," in *Proc. ACM SIGMETRICS Perform. Eval. Rev.*, 2015, vol. 43, pp. 191–204.

[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, Sep. 2012.

[20] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1119–1126.

[21] 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.

[22] 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.

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

[24] 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.

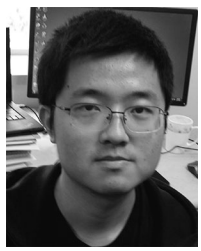
[25] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Process.*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.

[26] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A class of prediction-correction methods for time-varying convex optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591, Sep. 2016.

[27] A. Simonetto and E. DallAnese, "Prediction-correction algorithms for time-varying constrained optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5481–5494, Oct. 2017.

[28] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1973–1986, Jul. 2018.

[29] 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.

Since 2017, he has been a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. His research interests include optimization, game theory, signal processing, and probabilistic methods, with applications to information/social networks.

K. J. Ray Liu (F'03) is currently a Christine Kim Eminent Professor of information technology with the University of Maryland, College Park, MD, USA. He leads the Maryland Signals and Information Group conducting research encompassing broad areas of information and communications technology with recent focus on smart radios for smart life.

Prof. Liu was named a Distinguished Scholar-Teacher of the University of Maryland in 2007.