On the Time-Varying Constraints and Bandit Feedback of Online Convex Optimization

Xuanyu Cao and K. J. Ray Liu

Abstract—**In this paper, online convex optimization (OCO) 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 function feedback, in which complete information about the objective function and constraint functions is revealed to the agent 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 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 online algorithm is proposed and we also establish its sublinear expected regret and sublinear expected constraint violations. 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.**

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], e.g., smart grids with uncertain supply of renewable energy [3], [4] and data centers with uncertain user demands [5], [6]. Unlike traditional static optimization problems [7], [8], 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, complete or partial feedback information of the current objective/constraint functions is revealed to the agent. In [9], 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 time duration). The regret was further reduced to be $\mathcal{O}(\log T)$ by several online algorithms in [10]. While the offline benchmark was static in [9], [10], dynamic benchmarks were adopted in [11] and [12], where algorithms with sublinear regrets were presented. In [9]–[11], each time after an action is submitted, complete information about the objective function is revealed, i.e., the agent receives *function feedback*. This assumption is too restrictive for many applications in which only values of the objective function at the chosen action or several points near the action is available to the agent. This information scenario is called *bandit feedback*. Such a bandit version of OCO problem was studied in [13] for single point bandit feedback and in [14] for multi-point bandit feedback.

The aforementioned papers were concentrated on unconstrained OCO while many practical optimization problems invovled constraints. This discrepancy motivated several works on constrained OCO. In [15], constrained OCO with time-invariant constraints was studied by Mahdavi *et al.* and a corresponding continuous time version was considered by Paternain and Ribeiro in [16]. OCO with affine equality constraints was examined in [17] while distributed OCO problems over networks with consensus or proximity constraints were analyzed by Koppel *et al.* in [18] and [19], respectively. 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. Recently, constrained OCO with time-varying constraints was studied in [20]. There, function feedback (i.e., 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 a nonlinear optimization problem at each time instant. These limitations made the algorithm of [20] computationally inefficient and not suitable for bandit feedback.

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 function feedback and bandit feedback. Specifically, for constrained OCO with time-varying constraints and function feedback, we propose a computationally efficient online algorithm (Algorithm 1), which only involvs direct closed-form computations at each time. 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). For constrained OCO with time-varying constraints and bandit feedback, we propose an online algorithm (Algorithm 2) based on appropriate stochastic approximations of Algorithm 1. Sublinear expected regret and sublinear expected constraint violations are also demonstrated (Theorem 2). Two numerical examples, namely online quadratic programming and online logistic regression 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 function feedback and bandit feedback. This confirms the

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

theoretical guarantees in Theorem 1 and Theorem 2. The rest of this paper is organized as follows. In Section II, constrained OCO with time-varying constraints is formally formulated for both function feedback and bandit feedback. In Section III and Section IV, we propose and analyze algorithms for scenarios of function feedback and bandit feedback, respectively. Numerical experiments are presented in Section V, following which we conclude this work in Section VI.

II. Problem Formulation

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

A. Function Feedback

The constrained OCO problem with time-varying constraints is formulated as follows. At each time t, after the agent chooses an action $\mathbf{x}_t \in \mathcal{X}$, the nature will annouce not only a loss function $f_t(\cdot)$ but also a vector-valued constraint function $\mathbf{g}_t : \mathbb{R}^n \mapsto \mathbb{R}^m$ to the agent. Such an information scenario is called *function feedback* as the agent receives complete information about the loss function after the action is chosen. 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 revealed after the agent has already chosen the action $\mathbf{x}_t$. 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, e.g., smart grid in which the controller aims at balancing the energy supply and demand in the long run.

Thus, 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:

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

Solving problem (2) exactly is still impossible in the online setting here as the information about the loss functions and constraint functions are 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 $\texttt{Vio}^i(T) := \sum_{t=1}^T g_t^i(\mathbf{x}_t)$, $i = 1, ..., m$, where $g_t^i(\cdot)$ is the i-th component of vector-valued constraint function $\mathbf{g}_t(\cdot)$, i.e., $\mathbf{g}_t(\mathbf{x}) = [g_t^1(\mathbf{x}), ..., g_t^m(\mathbf{x})]^\intercal$. An ideal action sequence should possess sublinear regret and constraint violations, i.e., $\texttt{Reg}(T) \le o(T)$ and $\texttt{Vio}^i(T) \le o(T)$, $\forall i = 1, ..., m$. In such a case, 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, asymptotically, the average regret and the average constraint violations become non-positive as the time length T goes to infinity so that the performance of the sequence $\{\mathbf{x}_t\}$ is no worse than that of the benchmark sequence $\{\mathbf{x}_t^*\}$.

B. Bandit Feedback

In the previous subsection, we formulate the constrained OCO with function feedback, i.e., complete information about the loss function $f_t(\cdot)$ and the constraint function $\mathbf{g}_t(\cdot)$ 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 complete information regarding the functions $f_t(\cdot)$ and $\mathbf{g}_t(\cdot)$. Instead, the agent only knows the values of $f_t(\cdot), \mathbf{g}_t(\cdot)$ 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 $\mathbf{x}_t$ through accurate prediction and inference. The manager is not be aware of the complete information of the loss function $f_t(\cdot)$ because she does not know what her loss will be if she makes other arbitrary investment decisions. Similar arguments hold for the constraint function $\mathbf{g}_t(\cdot)$.

C. Assumptions and Definitions

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., $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_2 \le 1\}$ 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 $i = 1, ..., m$ and $t = 1, 2, ...$*

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 $\mathbf{x} \in \mathcal{X},\ i = 1, ..., m,\ t = 1, 2, ...$*

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 $\mathbf{x} \in \mathcal{X},\ t = 1, 2, ...$*

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 $t = 1, 2, ...$*

We further define the projection operator as follows, which will be used 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 $\Pi_{\mathcal{S}}(\mathbf{y})$.*

III. Constrained OCO with Function Feedback

In this section, we develop an online algorithm for the constrained OCO problem with function feedback. 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 4. Performance analysis indicates that the algorithm can achieve sublinear regret and sublinear constraint violations simultaneously.

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

$$\mathfrak{L}_t(\mathbf{x}, \boldsymbol{\lambda}) = f_t(\mathbf{x}) + \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{g}_t(\mathbf{x}) - \frac{\delta\eta}{2}\|\boldsymbol{\lambda}\|_2^2, \tag{3}$$

where $\boldsymbol{\lambda}$ is the Lagrangian multiplier; $\eta > 0$ is the stepsize of the algorithm to be used later; δ is some positive number to be determined by later analysis. 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 performs a primal descent step for the modified Lagrangian $\mathfrak{L}_t$ to obtain the new action $\mathbf{x}_{t+1}$:

$$\begin{aligned}\mathbf{x}_{t+1} &= \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta\nabla_{\mathbf{x}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)) && (4)\\ &= \Pi_{\mathcal{X}}\left(\mathbf{x}_t - \eta\left(\nabla f_t(\mathbf{x}_t) + \sum_{i=1}^{m}\lambda_t^i \nabla g_t^i(\mathbf{x}_t)\right)\right). && (5)\end{aligned}$$

In addition, the agent performs a dual ascent step for $\mathfrak{L}_t$ to compute the new dual variable $\boldsymbol{\lambda}_{t+1}$:

$$\begin{aligned}\boldsymbol{\lambda}_{t+1} &= \Pi_{\mathbb{R}_+^m}(\boldsymbol{\lambda}_t + \eta\nabla_{\boldsymbol{\lambda}}\mathfrak{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)) && (6)\\ &= \Pi_{\mathbb{R}_+^m}\left(\boldsymbol{\lambda}_t + \eta\left(\mathbf{g}_t(\mathbf{x}_t) - \delta\eta\boldsymbol{\lambda}_t\right)\right), && (7)\end{aligned}$$

where $\mathbb{R}_+^m$ denotes the non-negative orthant. Based on the updates specified in (5) and (7), we summarize the proposed algorithm for constrained OCO with function feedback in Algorithm 1. We note that the updates (5) and (7) only involve closed-form computations and do not need to solve any optimization problems, indicating high computational efficiency of Algorithm 1.

Algorithm 1 The algorithm for constrained OCO with function feedback

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

Next, 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 of the benchmark sequence $\{\mathbf{x}_t^*\}$ is sublinear. The formal definition of the drift is given in the following.

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

The main theorem regarding the performance of Algorithm 1 is presented as follows.

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:*

$$\begin{aligned}&\sum_{t=1}^{T}[f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]\\ &\leq \frac{5R^2}{2}\sqrt{\frac{T}{\Delta(T)}} + \left(R + \frac{m+1}{2}G^2 + D^2\right)\sqrt{T\Delta(T)}\\ &= \mathcal{O}\left(\sqrt{T\Delta(T)}\right),\end{aligned} \tag{8}$$

and for any $i = 1, ..., m$:

$$\begin{aligned}\sum_{t=1}^{T} g_t^i(\mathbf{x}_t) &\leq \sqrt{2\left(((m+1)G^2+1)\sqrt{T\Delta(T)} + \sqrt{\frac{T}{\Delta(T)}}\right)}\\ &\times\sqrt{FT + \frac{5R^2}{2}\sqrt{\frac{T}{\Delta(T)}} + \left(R + \frac{m+1}{2}G^2 + D^2\right)\sqrt{T\Delta(T)}} && (9)\\ &= \mathcal{O}\left(T^{\frac{3}{4}}\Delta(T)^{\frac{1}{4}}\right). && (10)\end{aligned}$$

Remark 1. *Since the drift sequence $\Delta(\cdot)$ is sublinear, (8) and (10) are both sublinear and so are the regret $\mathtt{Reg}(T) = \sum_{t=1}^T [f_t(\mathbf{x}_t) - f_t(\mathbf{x}_t^*)]$ and the constraint violations $\mathtt{Vio}^i(T) = \sum_{t=1}^T g_t^i(\mathbf{x}_t)$.*

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 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 instead of the complete information 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.

Before formally developing the algorithm, we first present some preliminaries regarding stochastic approximation of gradients. 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 [13], the following result regarding the gradient of $\hat{\phi}$ was shown.

Lemma 1. *Let $\mathbb{S}$ denote the unit Euclidean sphere in $\mathbb{R}^n$. Then, we have:*

$$\nabla\hat{\phi}(\mathbf{x}) = \frac{n}{\xi}\mathbb{E}_{\mathbf{u}\sim\mathcal{U}(\mathbb{S})}[\phi(\mathbf{x}+\xi\mathbf{u})\mathbf{u}]. \tag{11}$$

As such, a reasonable estimate of $\nabla\phi(\mathbf{x})$ is $\frac{n}{\xi}\phi(\mathbf{x}+\xi\mathbf{u})\mathbf{u}$, where $\mathbf{u}$ is some random vector uniformly distributed on $\mathbb{S}$. This approximation is useful in the later algorithm.

Define $\tilde{g}_t(\mathbf{x}) := \max_{i=1,...,m} g_t^i(\mathbf{x})$, $\hat{f}_t(\mathbf{x}) := \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[f_t(\mathbf{x}+\xi\mathbf{v})]$ and $\hat{g}_t(\mathbf{x}) := \mathbb{E}_{\mathbf{v}\sim\mathcal{U}(\mathbb{B})}[\tilde{g}_t(\mathbf{x}+\xi\mathbf{v})]$. The per-slot problem can be rewritten as $\min\{f_t(\mathbf{x})|\tilde{g}_t(\mathbf{x}) \le 0\}$. Define a modified Lagrangian associated with this problem:

$$\hat{\mathfrak{L}}_t(\mathbf{x},\lambda) := \hat{f}_t(\mathbf{x}) + \lambda\hat{g}_t(\mathbf{x}) - \frac{\eta\delta}{2}\lambda^2. \tag{12}$$

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:

$$\begin{aligned}&\nabla_{\mathbf{x}}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t) = \nabla\hat{f}_t(\mathbf{x}_t) + \lambda_t\nabla\hat{g}_t(\mathbf{x}_t),\\ &\frac{\partial}{\partial\lambda}\hat{\mathfrak{L}}_t(\mathbf{x}_t,\lambda_t) = \hat{g}_t(\mathbf{x}_t) - \eta\delta\lambda_t.\end{aligned}$$

According to Lemma 1, 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 $\frac{n}{2\xi}[f_t(\mathbf{x}_t + \xi\mathbf{u}_t) - f_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u}_t$. 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 $\frac{n}{2\xi}[\tilde{g}_t(\mathbf{x}_t + \xi\mathbf{u}_t) - \tilde{g}_t(\mathbf{x}_t - \xi\mathbf{u}_t)]\mathbf{u_t}$. 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:

$$\begin{aligned}\mathbf{p}_t =& \frac{n}{2\xi}[f_t(\mathbf{x}_t+\xi\mathbf{u}_t) - f_t(\mathbf{x}_t-\xi\mathbf{u}_t) + \lambda_t(\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t)\\ &- \tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t))]\mathbf{u}_t.\end{aligned} \tag{13}$$

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:

$$q_t = \frac{1}{2}[\tilde{g}_t(\mathbf{x}_t+\xi\mathbf{u}_t) + \tilde{g}_t(\mathbf{x}_t-\xi\mathbf{u}_t)] - \eta\delta\lambda_t. \tag{14}$$

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

$$\mathbf{x}_{t+1} = \Pi_{(1-\alpha)\mathcal{X}}(\mathbf{x}_t - \eta\mathbf{p}_t), \tag{15}$$

where $\alpha \in [\frac{\xi}{r}, 1)$. We initialize $\mathbf{x}_1 \in (1-\alpha)\mathcal{X}$ so that $\mathbf{x}_t \in (1-\alpha)\mathcal{X}$ for any $t = 1, 2, ...$ Thus, we 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. In addition, the agent updates the dual variable by an approximated dual ascent to obtain λ_{t+1}:

$$\lambda_{t+1} = \Pi_{\mathbb{R}^+}(\lambda_t + \eta q_t). \tag{16}$$

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, to operate Algorithm 2, the agent only needs bandit feedback information at two points ($\mathbf{x}_t \pm \xi\mathbf{u}_t$) very close to the action $\mathbf{x}_t$ instead of the complete information about the loss function and constraint functions.

Algorithm 2 The algorithm for constrained OCO with bandit feedback

```
Initialize $\mathbf{x}_1 \in (1-\alpha)\mathcal{X}$ and $\lambda_1 = 0$.
for $t = 1, 2, ..., T$ do
    Submit the action $\mathbf{x}_t$.
    Generate $\mathbf{u}_t$ according to a uniform distribution on the unit sphere $\mathbb{S}$.
    Query the values $f_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$ and $\tilde{g}_t(\mathbf{x}_t \pm \xi\mathbf{u}_t)$.
    Compute $\mathbf{p}_t$ and $q_t$ based on (13) and (14)
    Update the primal variable, i.e., the action, according to (15) to obtain the new action $\mathbf{x}_{t+1}$.
    Update the dual variable according to (16) to obtain the new dual variable $\lambda_{t+1}$.
end for
```

The main theorem regarding the expected regret and expected constraint violations is presented in the following.

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 $\frac{\Delta(T)}{T} \le \frac{1}{2(2n^2G^2+1)^2}$. Set $\xi = \frac{1}{T}$, $\alpha = \frac{1}{rT}$, $\delta = 2n^2G^2 + 1$ and $\eta = \sqrt{\frac{\Delta(T)}{T}}$. Then, we have:*

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} f_t(\mathbf{x}_t)\right] - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*)\\ &\le \left(R + n^2G^2 + D^2\right)\sqrt{T\Delta(T)}\\ &\quad + \left(\frac{5R^2}{2} + \frac{2GD}{2n^2G^2+1} + \frac{GD}{2n^2G^2+1}\left(\frac{R}{r}+1\right)\right)\sqrt{\frac{T}{\Delta(T)}}\\ &= \mathcal{O}\left(\sqrt{T\Delta(T)}\right),\end{aligned} \tag{17}$$

and for any $i = 1, ..., m$:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} g_t^i(\mathbf{x}_t)\right] \le \mathbb{E}\left[\sum_{t=1}^{T} \tilde{g}_t(\mathbf{x}_t)\right] \\
&\le 3G + \sqrt{2\left(\left(2n^2G^2+1\right)\sqrt{T\Delta(T)} + \sqrt{\frac{T}{\Delta(T)}}\right)} \\
&\quad \times \Bigg[FT + \left(R + n^2G^2 + D^2\right)\sqrt{T\Delta(T)} \\
&\quad + \left(\frac{5R^2}{2} + \frac{2GD}{2n^2G^2+1} + \frac{GD}{2n^2G^2+1}\left(\frac{R}{r}+1\right)\right)\sqrt{\frac{T}{\Delta(T)}} \\
&\quad + G\left(2 + \frac{R}{r}\right)\Bigg]^{\frac{1}{2}} \\
&= \mathcal{O}\left(T^{\frac{3}{4}}\Delta(T)^{\frac{1}{4}}\right). \qquad (18)
\end{aligned}$$

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

V. Numerical Experiments

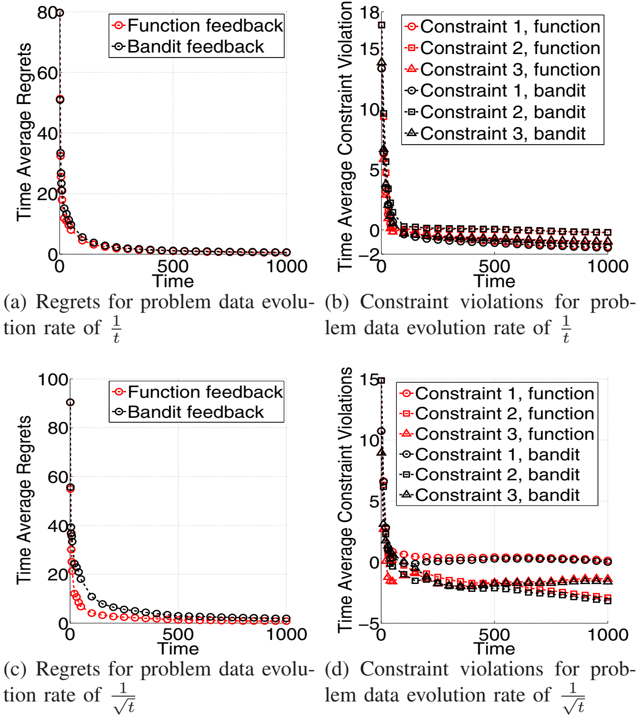
In this section, numerical experiments are conducted to corroborate the effectiveness of the proposed algorithms for constrained OCO with function feedback or bandit feedback.

First, we study a numerical example of online quadratic programming (OQP):

$$\text{Minimize}_{\mathbf{x}\in\mathcal{X}}\ \mathbf{x}^\mathsf{T}\mathbf{A}_t\mathbf{x} + \mathbf{b}_t^\mathsf{T}\mathbf{x}, \quad \text{subject to } \mathbf{C}_t\mathbf{x} \preceq \mathbf{d}_t, \quad (19)$$

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; $\mathbf{b}_t \in \mathbb{R}^n$, $\mathbf{C}_t \in \mathbb{R}^{m\times n}$, $\mathbf{d}_t \in \mathbb{R}^m$; $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x}\|_2 \le R\}$ is the action set with R some positive number. The problem (19) 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 $\mathbf{g}_t(\mathbf{x}_t) = \mathbf{C}_t\mathbf{x} - \mathbf{d}_t$. 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 $\mathbf{d}_t$. Such an OQP formulation has broad applications in many signal processing and control problems, such as dynamic resource allocation and dynamic linear regression. Roughly speaking, the time-varying problem data are generated such that each problem data at time $t+1$ is a perturbed version of the problem data at time t with perturbations uniformly distributed on $[-\frac{1}{2t}, \frac{1}{2t}]$. The parameters are set as: $m = 3, n = 10, T = 1000$, $R = 5$, $\eta = \frac{1}{\sqrt{T}}$, $\delta = 10$, $\xi = \frac{1}{T}$, $r = \frac{R}{2}$ and $\alpha = \frac{1}{rT}$.

We apply the proposed algorithms, i.e., Algorithm 1 and Algorithm 2, to the OQP. The time average regrets $\frac{\mathtt{Reg}(t)}{t}$ and the time average constraint violations $\frac{\mathtt{Vio}^i(t)}{t}$ are shown in Fig. 1-(a) and Fig. 1-(b), respectively. The scenarios of both function 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 Theorem 1 and Theorem 2. To investigate the impact of the evolution rate of the time-varying problem data, we alter the distribution of all the disturbances used in the problem data generation process to be uniform distribution over the interval $\left[-\frac{1}{2\sqrt{t}}, \frac{1}{2\sqrt{t}}\right]$, which increases the evolution speed of the problem data. The step size η is changed to be $0.2T^{-\frac{1}{4}}$ accordingly (as per specification in Theorems 1 and 2). The corresponding time average regrets and time average constraint violations are reported in Fig. 1-(c) and Fig. 1-(d), respectively. In Fig. 1-(c), we observe that the time average regrets are larger than that of the $\frac{1}{t}$ evolution rate (Fig. 1-(a)), especially for the scenario of bandit feedback. However, the time average regrets can still converge to zero, as guaranteed theoretically. Comparing Fig. 1-(d) with Fig. 1-(b), we remark that the constraint violations do not change much and are still negative as time approaches infinity.



(a) Regrets for problem data evolution rate of $\frac{1}{t}$

(b) Constraint violations for problem data evolution rate of $\frac{1}{t}$

(c) Regrets for problem data evolution rate of $\frac{1}{\sqrt{t}}$

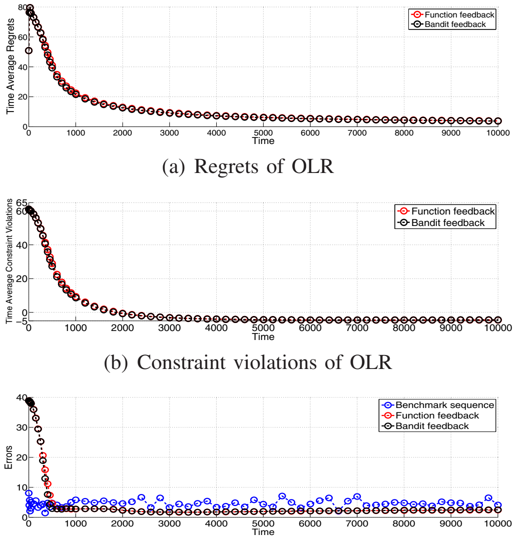
(d) Constraint violations for problem data evolution rate of $\frac{1}{\sqrt{t}}$

Fig. 1: Regrets and constraint violations for OQP with different problem data evolution rates.

Next, we examine a numerical example of online logistic regression (OLR):

$$\begin{aligned}
&\text{Minimize}_{\mathbf{x}\in\mathcal{X}}\ \sum_{i=1}^{k} \log\left(1 + \exp\left(-l_{i,t}\mathbf{u}_{i,t}^\mathsf{T}\mathbf{x}\right)\right) \\
&\qquad \text{subject to } \|\mathbf{x}\|_1 \le a_t,
\end{aligned} \qquad (20)$$

where $\mathbf{u}_{i,t} \in \mathbb{R}^n$ is the i-th training point at time t and $l_{i,t} \in \{-1, 1\}$ is the corresponding label. a_t is a threshold on the l_1 norm of the weight vector $\mathbf{x}$ to enforce sparsity. $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n | \|\mathbf{x}\|_\infty \le M\}$ is the action set and M is some positive number. Such an OLR problem (20) is in the form of (1) with $f_t(\mathbf{x}) = \sum_{i=1}^{k} \log\left(1 + \exp\left(-l_{i,t}\mathbf{u}_{i,t}^\mathsf{T}\mathbf{x}\right)\right)$ and $g_t(\mathbf{x}) = \|\mathbf{x}\|_1 - a_t$. When deciding $\mathbf{x}_t$, the agent does not know the



(a) Regrets of OLR

(b) Constraint violations of OLR

(c) Errors with respect to the truth

Fig. 2: Regrets, constraint violations and errors with respect to the true weight vectors for OLR

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 recursively with an auxiliary ground truth weight vector sequence $\{\mathbf{x}_t^{\#}\}$. The parameters of the problem are $n = 5, k = 20, T = 10000, \eta = 0.2\frac{1}{\sqrt{T}}, \delta = 10, \xi = \frac{1}{T}, r = M = 30, \alpha = \frac{1}{rT}$.

We apply the proposed algorithms, i.e., Algorithm 1 and Algorithm 2, to the OLR 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. 2-(a) and Fig. 2-(b), respectively. We remark that, for both function 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. 2-(c), we plot the tracking errors of function feedback ($\|\mathbf{x}_t - \mathbf{x}_t^{\#}\|_2$, where $\mathbf{x}_t$ is generated by Algorithm 1 with function 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 (20), i.e., the benchmark or the posteriori optima). From Fig. 2-(c), we observe that, for both function 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 examine constrained online convex optimization (OCO) problems with time-varying constraints. For function feedback, we propose a computationally efficient online algorithm and establish its sublinear regret and constraint violations under the assumption that the drift of the benchmark sequence is sublinear. Moreover, we investigate a bandit version of the constrained OCO problem and propose an online algorithm for bandit feedback. The corresponding sublinear results for the expected regrets and the expected constraint violations are also demonstrated. Finally, numerical examples are presented to validate the effectiveness of the proposed algorithms.

References

[1] E. Hazan *et al.*, “Introduction to online convex optimization,” *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[2] S. Shalev-Shwartz *et al.*, “Online learning and online convex optimization,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[3] Y. Zhang, M. Hajiesmaili, S. Cai, M. Chen, and Q. Zhu, “Peak-aware online economic dispatching for microgrids,” *IEEE Transactions on Smart Grid*, 2016.

[4] 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 *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS ’13, (New York, NY, USA), pp. 53–66, ACM, 2013.

[5] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, “Dynamic right-sizing for power-proportional data centers,” *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

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

[7] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[8] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[9] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” *International Conference on Machine Learning*, 2003.

[10] E. Hazan, A. Agarwal, and S. Kale, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[11] O. Besbes, Y. Gur, and A. Zeevi, “Non-stationary stochastic optimization,” *Operations Research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[12] E. C. Hall and R. M. Willett, “Online convex optimization in dynamic environments,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[13] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, “Online convex optimization in the bandit setting: Gradient descent without a gradient,” in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’05, (Philadelphia, PA, USA), pp. 385–394, Society for Industrial and Applied Mathematics, 2005.

[14] A. Agarwal, O. Dekel, and L. Xiao, “Optimal algorithms for online convex optimization with multi-point bandit feedback.,” in *COLT*, pp. 28–40, Citeseer, 2010.

[15] M. Mahdavi, R. Jin, and T. Yang, “Trading regret for efficiency: online convex optimization with long term constraints,” *Journal of Machine Learning Research*, vol. 13, no. Sep, pp. 2503–2528, 2012.

[16] S. Paternain and A. Ribeiro, “Online learning of feasible strategies in unknown environments,” *IEEE Transactions on Automatic Control*, 2017.

[17] H. Wang and A. Banerjee, “Online alternating direction method,” *International Conference on Machine Learning (ICML)*, 2012.

[18] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, “A saddle point algorithm for networked online convex optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[19] A. Koppel, B. Sadler, and A. Ribeiro, “Proximity without consensus in online multi-agent optimization,” *IEEE Transactions on Signal Processing*, 2017.

[20] T. Chen, Q. Ling, and G. B. Giannakis, “An online convex optimization approach to proactive network resource allocation,” *IEEE Transactions on Signal Processing*, 2017.