2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Safe Online Convex Optimization with First-order Feedback

Spencer Hutchinson and Mahnoosh Alizadeh

***Abstract*— We study an online convex optimization problem where the player must satisfy an unknown constraint at all rounds, while only observing the gradient and function value of the constraint at the chosen actions. For this problem, we develop an algorithm that uses an optimistic set, which overestimates the constraint, to identify low-regret actions while using a pessimistic set, which underestimates the constraint, to ensure constraint satisfaction. Our analysis shows that this algorithm satisfies the constraint at all rounds while enjoying $\mathcal{O}(\sqrt{T})$ regret when the constraint function is smooth and strongly convex. We then extend our algorithm to a setting with time-varying constraints and prove that it enjoys similar guarantees in this setting. Lastly, we demonstrate the effectiveness of our algorithm with a set of numerical experiments.**

I. Introduction

The online convex optimization (OCO) setting, due to [1], is a sequential decision-making problem where a player chooses a vector action x_t at each round, and subsequently observes the loss function f_t and suffers the loss $f_t(x_t)$. This setting has received considerable attention due to its broad applicability to fields ranging from online advertising [2], [3] to network resource allocation [4], [5] and power systems [6], [7].

In the conventional OCO setting, the constraints on the player's actions are assumed to be entirely known. However, such constraints are often unknown in the real world, motivating various works with unknown constraints that either ensure constraint violation grows sublinearly [8], [9], [10], or that constraints are never violated [11], [12]. This work falls into the latter category in that we want to ensure the constraints are never violated, despite providing the player with limited information about them. Ensuring that constraints are never violated is of utmost importance in safety-critical applications, such as power systems and clinical trials, where constraint violations could result in serious consequences, such as infrastructure damage or patient harm.

In particular, we study a setting where the player needs to satisfy a fixed, but unknown, constraint and receives feedback of the constraint function value and gradient at each action that she plays. Note that this type of constraint feedback (first-order feedback) is often used in OCO problems with constraints, e.g. [13], [14], although such works do not ensure constraint satisfaction in every round (as we do). To address our stated problem, we propose the algorithm ROGD, which leverages both sets that underestimate the constraint set (which we call *pessimistic sets*) and sets that overestimate the constraint set (which we call *optimistic sets*) to efficiently balance the competing objectives of ensuring low regret and maintaining constraint satisfaction. Our analysis shows that this algorithm enjoys $\mathcal{O}(\sqrt{T})$ regret when the constraint function is smooth and strongly convex. Furthermore, we find that this algorithmic approach yields similar guarantees in a setting where the constraints are allowed to vary with time. The efficacy of our approach is then demonstrated empirically via a series of numerical experiments.

S. Hutchinson and M. Alizadeh are with Dept. of ECE, UCSB, Santa Barbara, CA, USA. This work is supported by NSF grant #1847096. E-mails: shutchinson@ucsb.edu, alizadeh@ucsb.edu

A. Related work

There is a large body of literature that study OCO with time-varying constraints. In particular, [8], [13], [15], [16], [17], [10], [18] study settings in which the constraints for a given round are adversarily chosen after the player chooses an action. In a related direction, [14], [18] studied a setting where the constraint functions are sampled iid. These works generally employ primal-dual methods to ensure that both the regret and the cumulative constraint violation grow sublinearly. This differs from our approach, where we aim to ensure that a fixed constraint is never violated, while providing limited feedback to the player.

Another related area of research is projection-free OCO, which aims to develop algorithms that do not require the costly projection operation that is required by standard OCO algorithms. One prominent direction in projection-free OCO aims to develop alternatives to projection that can be used with standard algorithms [19], [20]. Another direction in this field is focused on developing algorithms that use the cheaper linear optimization oracle [21], [22], [23]. A third direction in projection-free OCO avoids projections by allowing some constraint violation [9], [24], [10]. Even though some of these methods ensure that constraints are always satisfied without access to a projection oracle, they still assume access to some other oracle that uses the constraint, i.e. linear optimization oracle [21], [22], [23], membership oracle [20], or gradient and value of constraint function at any point [19]. This differs from our setting in which the player only receives feedback at the points that are played.

Most relevantly, there have also been various works that study OCO and other learning problems with unknown constraints that always need to be satisfied. In particular, [11] studies a safe OCO problem with an unknown linear constraint which the player receives noisy zero-order feedback of, and proposes an algorithm that first performs an iid exploration phase and then online gradient descent to get $\tilde{\mathcal{O}}(dT^{2/3})$ regret. This approach is then extended to distributed settings with convex and nonconvex cost functions in [12]. Our setting differs from [11], [12] in

979-8-3503-8265-5/$31.00 ©2024 AACC

that we consider nonlinear (smooth and strongly convex) constraints and provide the player with first-order feedback. Our algorithmic approach also differs in that our algorithm does not use a dedicated pure exploration phase to learn the constraint but instead uses an action-selection rule that automatically balances regret minimization and constraint satisfaction.

Other related safe learning works are [25], which studies an OCO problem where the cumulative loss at each round needs to stay below a threshold, [26], [27], which study a stochastic linear bandit problem with unknown constraints and noisy feedback, and [28], [29], [30], which study zero-order optimization where constraints need to always be satisfied. Although these works address similar challenges as we do, i.e. ensuring constraint satisfaction under uncertainty, the underlying problem differs and therefore our setting requires different methods.

B. Paper organization

We specify the problem setting in Section II, propose an algorithm for this problem in Section III and then analyze this algorithm in Section IV. In Section V, we extend our approach to a setting with time-varying constraints. We then provide numerical experiments in Section VI and concluding remarks in Section VII.

C. Notation

We use $\mathcal{O}(\cdot)$ to refer to big-O notation. Also, we denote the 2-norm by $\|\cdot\|$. For a natural number n, we use $[n]$ for the set $\{1, 2, ..., n\}$. For a matrix M, we use $M^\top$ to denote the transpose of M. A set $\mathcal{X} \subseteq \mathbb{R}^d$ is referred to as *convex* if $(1-\lambda)x + \lambda y \in \mathcal{X}$ for all $x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$. For a convex set $\mathcal{X}$, a function $f : \mathcal{X} \to \mathbb{R}$ is referred to as *convex* if $f((1-\lambda)x + \lambda y) \leq (1-\lambda)f(x) + \lambda f(y)$ for all $x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$. Also for a closed convex set $\mathcal{X} \subseteq \mathbb{R}^d$ and a vector $x \in \mathbb{R}^d$, we denote the projection operation with $\Pi_{\mathcal{X}}(y) = \arg\min_{x \in \mathcal{X}} \|x - y\|$. A useful fact is that for a closed convex set $\mathcal{X} \subseteq \mathbb{R}^d$ and vectors $y \in \mathbb{R}^d$ and $x \in \mathcal{X}$, it holds that $\|y - x\| \geq \|\Pi_{\mathcal{X}}(y) - x\|$.

II. Problem setup

We study an online convex optimization problem with an unknown constraint $\mathcal{G} = \{x \in \mathbb{R}^d : g(x) \leq 0\}$, where g is a convex function. This problem can be viewed as an iterative game between a player and an adversary, where at each round $t \in [T]$:

1) player chooses an action $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$,
2) adversary chooses f_t and player suffers cost $f_t(x_t)$,
3) player observes $\nabla f_t(x_t)$, $g(x_t)$ and $\nabla g(x_t)$.[1]

Critically, the player must ensure that $x_t \in \mathcal{G}$ for all $t \in [T]$ despite the fact that $\mathcal{G}$ is initially unknown. We take the *action set* $\mathcal{X}$ to be convex and closed, and refer to the *feasible set* as $\mathcal{Y} = \mathcal{X} \cap \mathcal{G}$. Furthermore, we only allow the adversary to choose *cost functions* of the form $f_t : \mathcal{X} \to \mathbb{R}$ that are differentiable and convex, and take the *constraint function* $g : \mathcal{X} \to \mathbb{R}$ to also be differentiable and convex.

Algorithm 1: Restrained Online Gradient Descent (ROGD)

Input: $\mathcal{X}, \eta, L, M$.
Set $\tilde{x}_1 = \mathbf{0}$ and $x_1 = \mathbf{0}$.
for $t = 1$ **to** T **do**
  Play x_t and observe $\nabla f_t(x_t), g(x_t), \nabla g(x_t)$.
  Update $\mathcal{Y}_t^o$ and $\mathcal{Y}_t^p$ with (1) and (2).
  $\tilde{x}_{t+1} = \Pi_{\mathcal{Y}_t^o}(\tilde{x}_t - \eta \nabla f_t(x_t))$.
  $\gamma_t = \max\{\mu \in [0, 1] : x_t + \mu(\tilde{x}_{t+1} - x_t) \in \mathcal{Y}_t^p\}$.[3]
  $x_{t+1} = x_t + \gamma_t(\tilde{x}_{t+1} - x_t)$.
end

In addition to ensuring constraint satisfaction, the player also aims to minimize her loss compared to the best action in hindsight. Concretely, the player aims to minimize her *static regret*, which is defined as

$$R_T^s := \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x_*),$$

where $x_* = \arg\min_{x \in \mathcal{Y}} \sum_{t=1}^{T} f_t(x)$. We use the following assumptions.

Assumption 1 (Bounded gradients): For all $x \in \mathcal{X}$ and $t \in [T]$, it holds that $\|\nabla f_t(x)\| \leq G$.

Assumption 2 (Bounded action set): There exists a positive real D such that $\|x - y\| \leq D$ for all $x, y \in \mathcal{X}$.

Assumption 3 (Initial feasible point): It holds that $\mathbf{0}$ is in $\mathcal{X}$ and $g(\mathbf{0}) \leq 0$.

Assumption 4 (Smooth and strongly convex constraint): The constraint function g is L-smooth and M-strongly convex on the set $\mathcal{X}$. That is, it holds for all $x, y \in \mathcal{X}$ that

$$g(y) \geq g(x) + \nabla g(x)^\top (y - x) + \frac{M}{2}\|y - x\|^2,$$
$$g(y) \leq g(x) + \nabla g(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2,$$

where $\kappa := L/M > 1$.[2]

Assumptions 1 and 2 are standard in the OCO setting, e.g. [1]. Assumption 3 ensures that there is a feasible point that is initially known by the player, which is necessary to ensure constraint satisfaction in the first round. Assumption 4 specifies that the constraint function is smooth and strongly convex, which is critical to our approach as it allows our algorithm to construct spherical sets that tightly overestimate and underestimate the constraint, respectively. This is discussed further in the next section.

III. Algorithm

To address the stated problem, we propose the algorithm Restrained Online Gradient Descent (ROGD) given in Al-

[1] In many formulations of the OCO problem, the player is given access to the entire cost function f_t rather than just the gradient at the chosen action $\nabla f_t(x_t)$. However, our algorithm (and many common OCO algorithms) only require access to $\nabla f_t(x_t)$, so we formulate our problem as such.

[2] Assumption 4 implies that the constraint set $\mathcal{G}$ is smooth and strongly convex. Smooth action sets have been used for projection-free OCO [19] and strongly convex action sets have been used to prove faster rates for the Frank-Wolfe method in (offline) convex optimization [31].

gorithm 1. This algorithm maintains a *pessimistic set* ($\mathcal{Y}_t^p$), which is known to be a subset of the true feasible set ($\mathcal{Y}$), and an *optimistic set* ($\mathcal{Y}_t^o$), which is known to be a superset of the true feasible set. In each round, the algorithm updates the *optimistic action* ($\tilde{x}_t$) with a projected gradient descent step on the optimistic set (line 5), and then moves the *played action* (x_t) as far as possible towards the optimistic action while staying within the pessimistic action set (lines 6 and 7).

Intuitively, the optimistic set is used to guide the algorithm towards low-regret (but potentially unsafe) actions, while the pessimistic set is used to ensure that the played actions do in fact satisfy the constraints. Specifically, the optimistic action is updated with projected gradient descent on the optimistic set and so, given the analysis of the classical projected gradient descent algorithm [1] and the fact that the optimistic set contains the true feasible set, we know that the regret due to the optimistic action will be low (i.e. $\mathcal{O}(\sqrt{T})$). However, the optimistic action might not satisfy the constraint so we cannot play the optimistic action. We instead play an action that is in the pessimistic set, ensuring constraint satisfaction, and is as close as possible to the optimistic action. Due to the construction of the optimistic and pessimistic sets (as discussed next) and with a step size η that shrinks with T (e.g. $\eta \approx 1/\sqrt{T}$), this approach ensures that the played actions stay near to the optimistic actions. As a result, the regret of the played actions will also be low.

The optimistic and pessimistic sets are constructed using the strong convexity and smoothness of the constraint function which is assured by Assumption 4. In particular, the optimistic and pessimistic action sets are defined as

$$\mathcal{Y}_t^o := \Big\{ x \in \mathcal{X} : g(x_t) + \nabla g(x_t)^\top (x - x_t) + \frac{M}{2}\|x - x_t\|^2 \leq 0 \Big\}, \tag{1}$$

and,

$$\mathcal{Y}_t^p := \Big\{ x \in \mathcal{X} : g(x_t) + \nabla g(x_t)^\top (x - x_t) + \frac{L}{2}\|x - x_t\|^2 \leq 0 \Big\} \tag{2}$$

respectively. It follows from these definitions that $\mathcal{Y}_t^p \subseteq \mathcal{Y} \subseteq \mathcal{Y}_t^o$. Therefore, it holds that x_t is in $\mathcal{Y}$ for all t given that x_t is chosen to be in $\mathcal{Y}_{t-1}^p$.

IV. Regret analysis

In this section, we prove an upper bound on the static regret of the proposed algorithm ROGD. The following theorem shows that, after T rounds, ROGD enjoys static regret less than $\kappa DG\sqrt{T}$ with an appropriate choice of step size η. Despite the fact that the constraint is unknown in this setting, the static regret bound of ROGD only differs from the static regret bound of standard online gradient descent with known constraints by a factor of κ, which is the condition number of the constraint function.

Theorem 1: Let Assumptions 1, 2, 3 and 4 hold. The static regret of ROGD (Algorithm 1) satisfies

$$R_T^s \leq \left(\kappa - \frac{1}{2}\right) G^2 \eta T + \frac{D^2}{2\eta}.$$

Choosing $\eta = \frac{D}{G\sqrt{T}}$ ensures that $R_T^s \leq \kappa DG\sqrt{T}$.

In the following subsections, we first provide the supporting lemmas and then give the proof of Theorem 1.

A. Supporting lemmas

There are three key lemmas that are needed to analyze the performance of ROGD. The first lemma shows that the scaling on the update of the played action (i.e. γ_t in line 6) is lower bounded by a constant.

Lemma 1: Let Assumptions 3 and 4 hold. Then, we have that $\gamma_t \geq 1/\kappa$ for all $t \in [T]$.

Proof: Let $y := \tilde{x}_{t+1} - x_t$. Since $\tilde{x}_{t+1}$ is in $\mathcal{Y}_t^o$ by definition, we know that

$$\begin{aligned} &g(x_t) + \nabla g(x_t)^\top y + \frac{M}{2}\|y\|^2 \leq 0 \\ \iff &\nabla g(x_t)^\top y + \frac{M}{2}\|y\|^2 \leq -g(x_t). \end{aligned}$$

Then, we aim to find an $\alpha \in [0,1]$ such that $u = x_t + \alpha(\tilde{x}_{t+1} - x_t) = x_t + \alpha y$ is in $\mathcal{Y}_t^p$. Due to the convexity of $\mathcal{X}$ and the fact that x_t and $\tilde{x}_{t+1}$ are in $\mathcal{X}$, we know that u is in $\mathcal{X}$ for any such α. Choosing $\alpha = 1/\kappa$, we have that

$$\begin{aligned} &g(x_t) + \nabla g(x_t)^\top (u - x_t) + \frac{L}{2}\|u - x_t\|^2 \\ &= g(x_t) + \alpha \nabla g(x_t)^\top y + \alpha^2 \frac{L}{2}\|y\|^2 \\ &= g(x_t) + \alpha\left(\nabla g(x_t)^\top y + \alpha\frac{L}{2}\|y\|^2\right) \\ &= g(x_t) + \alpha\left(\nabla g(x_t)^\top y + \frac{M}{2}\|y\|^2\right) \\ &\leq g(x_t) - \alpha g(x_t) \\ &= (1-\alpha) g(x_t) \leq 0, \end{aligned}$$

where the last inequality follows from the fact that x_t is in $\mathcal{Y}$ for all t and therefore $g(x_t) \leq 0$. Since $u = x_t + \alpha(\tilde{x}_{t+1} - x_t)$ is in $\mathcal{Y}_t^p$ with $\alpha = 1/\kappa$ and γ_t is defined as the largest such α, we know that $\gamma_t \geq 1/\kappa$ by definition. ∎

We then use Lemma 1 to show that, with an appropriate choice of step size, the distance between the optimistic and played actions is always bounded by a constant.

Lemma 2: Let Assumptions 1, 3 and 4 hold. Fix any $\epsilon > 0$. If $\eta = \frac{1/\kappa}{G(1-1/\kappa)}\epsilon$, then it holds that $\|x_t - \tilde{x}_t\| \leq \epsilon$ for all t.

Proof: We show this by induction. The base case holds by definition as $\tilde{x}_1 = x_1 = \mathbf{0}$. Suppose that $\|x_t - \tilde{x}_t\| \leq \epsilon$,

[3]Note that the update of γ_t in line 6 is always well-defined in the sense that there exists a $\mu \in [0,1]$ such that $x_t + \mu(\tilde{x}_{t+1} - x_t) \in \mathcal{Y}_t^p$ for every $t \in [T]$. To see this, first note that if x_t is in $\mathcal{Y}$ then $g(x_t) \leq 0$ and therefore, x_t is in $\mathcal{Y}_t^p$ and choosing $\mu = 0$ ensures that $x_{t+1} = x_t + \mu(\tilde{x}_{t+1} - x_t) \in \mathcal{Y}_t^p \subseteq \mathcal{Y}$. Since $x_1 \in \mathcal{Y}$ by definition, it follows by induction over t that γ_t is well-defined for all $t \in [T]$.

then we have that

$$\begin{aligned}
\|\tilde{x}_{t+1} - x_{t+1}\| &= \|\tilde{x}_{t+1} - x_t - \gamma_t(\tilde{x}_{t+1} - x_t)\| \\
&= (1-\gamma_t)\|\tilde{x}_{t+1} - x_t\| \\
&\le (1-1/\kappa)\|\tilde{x}_{t+1} - x_t\| && \text{(a)} \\
&= (1-1/\kappa)\|\Pi_{\mathcal{Y}_t^o}(\tilde{x}_t - \eta\nabla f_t(x_t)) - x_t\| \\
&\le (1-1/\kappa)\|\tilde{x}_t - \eta\nabla f_t(x_t) - x_t\| && \text{(b)} \\
&\le (1-1/\kappa)(\|\tilde{x}_t - x_t\| + \eta\|\nabla f_t(x_t)\|) && \text{(c)} \\
&\le (1-1/\kappa)(\epsilon + \eta G) && \text{(d)} \\
&= (1-1/\kappa)\left(\epsilon + \frac{1/\kappa}{G(1-1/\kappa)}\epsilon G\right) \\
&= \epsilon,
\end{aligned}$$

where (a) follows from Lemma 1, (b) follows from the fact that x_t is in $\mathcal{Y}_t^o$, (c) is the triangle inequality and (d) uses the induction hypothesis. ∎

The last of the technical lemmas, given in the following, provides a bound on the linearized loss at each round with respect to an arbitrary point in the feasible set. The lemma follows from the fact that the optimistic action is an overestimate of the feasible set and therefore a projection onto the optimistic set (as used by the algorithm in line 5) will shrink the distance to any feasible action.

Lemma 3: Let Assumptions 1 and 4 hold. Then, for any $v \in \mathcal{Y}$, it holds that

$$\begin{aligned}
&\nabla f_t(x_t)^\top(\tilde{x}_t - v) \\
&\quad\le \frac{1}{2\eta}\left(\|\tilde{x}_t - v\|^2 - \|\tilde{x}_{t+1} - v\|^2\right) + \frac{1}{2}\eta G^2,
\end{aligned}$$

for all $t \in [T]$.

Proof: Because $v \in \mathcal{Y} \subseteq \mathcal{Y}_t^o$, we know that

$$\begin{aligned}
&\|\tilde{x}_{t+1} - v\|^2 \\
&= \|\Pi_{\mathcal{Y}_t^o}(\tilde{x}_t - \eta\nabla f_t(x_t)) - v\|^2 \\
&\le \|\tilde{x}_t - \eta\nabla f_t(x_t) - v\|^2 \\
&= \|\tilde{x}_t - v\|^2 - 2\eta\nabla f_t(x_t)^\top(\tilde{x}_t - v) + \eta^2\|\nabla f_t(x_t)\|^2 \\
&\le \|\tilde{x}_t - v\|^2 - 2\eta\nabla f_t(x_t)^\top(\tilde{x}_t - v) + \eta^2 G^2.
\end{aligned}$$

The proof is complete by rearranging the last line and dividing by 2η. ∎

B. Proof of Theorem 1

Leveraging Lemmas 1, 2 and 3, we prove Theorem 1 as follows.

Proof: Since x_* is in $\mathcal{Y}$, we can use Lemma 3 with $v \leftarrow x_*$ and sum over t to get

$$\begin{aligned}
&\sum_{t=1}^{T}\nabla f_t(x_t)^\top(\tilde{x}_t - x_*) \\
&\le \frac{1}{2\eta}\sum_{t=1}^{T}(\|\tilde{x}_t - x_*\|^2 - \|\tilde{x}_{t+1} - x_*\|^2) + \frac{1}{2}G^2\eta T \qquad (3) \\
&= \frac{1}{2\eta}(\|\tilde{x}_1 - x_*\|^2 - \|\tilde{x}_{T+1} - x_*\|^2) + \frac{1}{2}G^2\eta T \\
&\le \frac{1}{2\eta}D^2 + \frac{1}{2}G^2\eta T.
\end{aligned}$$

Then, we can bound the static regret directly as

$$\begin{aligned}
R_T^s &= \sum_{t=1}^{T}(f_t(x_t) - f_t(x_*)) \\
&\le \sum_{t=1}^{T}\nabla f_t(x_t)^\top(x_t - x_*) && \text{(a)} \\
&= \sum_{t=1}^{T}\nabla f_t(x_t)^\top(x_t - \tilde{x}_t) + \sum_{t=1}^{T}\nabla f_t(x_t)^\top(\tilde{x}_t - x_*) \\
&\le G\sum_{t=1}^{T}\|x_t - \tilde{x}_t\| + \sum_{t=1}^{T}\nabla f_t(x_t)^\top(\tilde{x}_t - x_*) && \text{(b)} \\
&\le \frac{G^2(1-1/\kappa)\eta T}{1/\kappa} + \frac{D^2}{2\eta} + \frac{1}{2}G^2\eta T && \text{(c)} \\
&= \left(\kappa - \frac{1}{2}\right)G^2\eta T + \frac{D^2}{2\eta} \\
&= \kappa D G\sqrt{T}, && \text{(d)}
\end{aligned}$$

where (a) is due to the convexity of f_t, (b) is due to Cauchy-Schwarz and Assumption 1, (c) follows from applying Lemma 2 to the first term and (3) to the second term, and (d) uses the choice of step size $\eta = \frac{D}{G\sqrt{T}}$. ∎

V. Extension to time-varying constraints

In this section, we extend our algorithm and analysis to the setting where the constraints vary in each round. In particular, we consider the time-varying constraint $\mathcal{G}_t = \{x \in \mathbb{R}^d : g_t(x) \le 0\}$ with the time-varying constraint function g_t, where the constraint sets are monotone, i.e. $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \ldots \subseteq \mathcal{G}_T$. We also give the player feedback on the constraint for the next round such that, in each round $t \in [T]$, the player observes the feedback on the constraint for round $t+1$, i.e. $g_{t+1}(x_t)$ and $\nabla g_{t+1}(x_t)$.[4] In this setting, the player must ensure that $x_t \in \mathcal{G}_t$ for all $t \in [T]$. We refer to the feasible set in round t as $\mathcal{Y}_t := \mathcal{X} \cap \mathcal{G}_t$.

Since the feasible set varies in each round, the notion of static regret used in the original setting is ill-defined in this setting. Instead, we measure the performance of the player against the best action at each round, which is known as *dynamic regret*. That is,

$$R_T^d := \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x_t^*)$$

where $x_t^* = \arg\min_{x \in \mathcal{Y}_t} f_t(x)$.

We directly use Assumptions 1 and 2 from the original setting and assume that the constraints at all time steps satisfy Assumptions 3 and 4.

Remark 1: Our setting differs from most existing works on OCO with time-varying constraints, e.g. [13], [30], in that we consider 1) monotone constraint sets, 2) feedback on the next constraint, 3) no constraint violation, and 4) regret compared to the best action in the feasible set at each

[4]This type of feedback can be considered a "prediction" of future constraints. Various types of predictions have been considered in the OCO setting, e.g. [32], [33].

round. Instead, existing works often consider 1) arbitrarily varying constraints, 2) feedback on the constraint in the current round, 3) sublinear constraint violation and 4) regret compared to the best action that satisfies the constraint in all rounds (referred to as the common feasible set).

A. Algorithm

In this section, we adapt ROGD (Algorithm 1) to the setting with time-varying constraints. To do so, we need to modify the algorithm to ensure that the optimistic set overestimates the true feasible set and that the pessimistic set underestimates the true feasible set. Specifically, we redefine the optimistic and pessimistic sets as

$$\mathcal{Y}_t^o := \Big\{x \in \mathcal{X} : g_{t+1}(x_t) + \nabla g_{t+1}(x_t)^\top (x - x_t) + \frac{M}{2}\|x - x_t\|^2 \leq 0\Big\},$$

and,

$$\mathcal{Y}_t^p := \Big\{x \in \mathcal{X} : g_{t+1}(x_t) + \nabla g_{t+1}(x_t)^\top (x - x_t) + \frac{L}{2}\|x - x_t\|^2 \leq 0\Big\}$$

respectively. Then, it follows from the strong convexity and smoothness of the constraint function that $\mathcal{Y}_t^p \subseteq \mathcal{Y}_{t+1} \subseteq \mathcal{Y}_t^o$ for all $t \in [T]$. Also, since $x_t \in \mathcal{Y}_{t-1}^p$ and the constraint sets are monotone, it holds that $x_t \in \mathcal{Y}_t \subseteq \mathcal{Y}_{t+1} \subseteq ... \subseteq \mathcal{Y}_T$.

B. Regret analysis

In this section, we give dynamic regret bounds for ROGD in the setting with time-varying constraints. As is typical in dynamic regret analysis, e.g. [1], we use the path length of the optimal actions as defined in the following.

Definition 1: The path length of the optimal actions $(x_t^*)_{t\in[T]}$ is defined as

$$P_T := \sum_{t=1}^{T-1} \|x_t^* - x_{t+1}^*\|.$$

With this, we then give the dynamic regret guarantees of ROGD as follows.

Theorem 2: The dynamic regret of ROGD (Algorithm 1) in the setting with time-varying constraints satisfies

$$R_T^d \leq \left(\kappa - \frac{1}{2}\right) G^2 \eta T + \frac{1}{\eta} D^2 + \frac{1}{\eta} D P_T.$$

In particular, choosing $\eta = \sqrt{(P_T + 1)/T}$ ensures that R_T^d is $\mathcal{O}(\sqrt{T(P_T + 1)})$ where we use $\mathcal{O}(\cdot)$ to hide all problem parameters except P_T and T.

We give the proof of Theorem 2 in Appendix A. This proof follows by extending Lemmas 1, 2 and 3 to this setting and then by bounding the regret in terms of the path length as done in [1].

Remark 2: In order for the regret bound in Theorem 2 to yield $\mathcal{O}(\sqrt{T(P_T + 1)})$ regret, the path length P_T needs to be known when choosing the step size η. In some applications, the path length may not be known in advance, so we leave it as future work to remove this requirement.

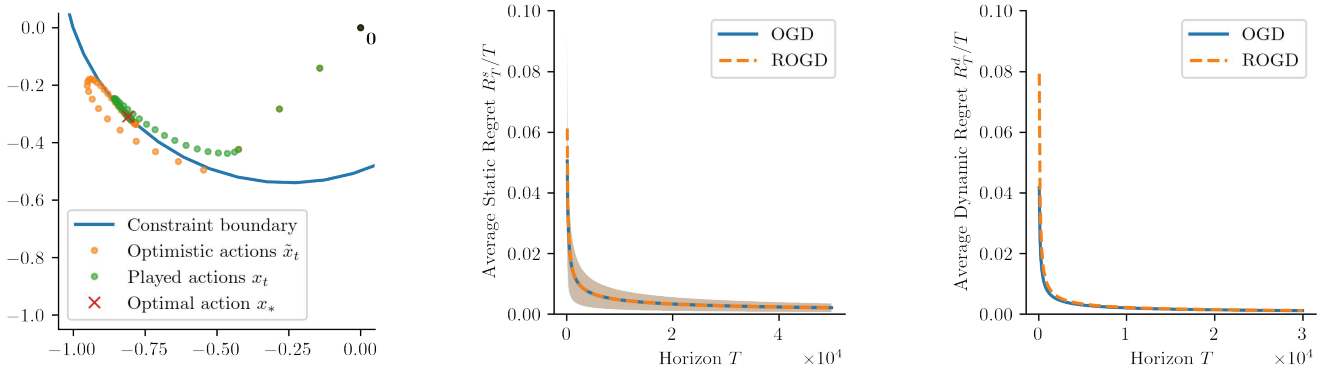
VI. Numerical experiments

In order to validate the theoretical results and illustrate the operation of ROGD, we give some numerical results as shown in Figure 1. We consider three different types of settings, fixed cost functions and constraint (Figures 1a), time-varying cost functions and fixed constraint (Figure 1b), and time-varying cost functions and constraints (Figure 1c). In the following, we provide the details on each of these settings.

For the setting with fixed cost functions and constraints (Figure 1a), we consider a linear cost function and quadratic constraint. In particular, we take the cost function to be $f_t(x) = f(x) = [1\ 1]x$ for all t, the constraint function to be $g(x) = 4\|x\|^2 + [2\ -2]x - 2$ and the action set to be $\mathcal{X} = \mathbb{B}$ where $d = 2$. We give the algorithm the information that the constraint function is 1-strongly convex and 8-smooth. We run ROGD with $T = 100$ and $\eta = \frac{D}{G\sqrt{T}}$ where $D = 2$ and $G = \sqrt{2}$, and plot the optimistic actions $\tilde{x}_t$ and the played actions x_t in Figure 1a. From this plot, we can see that the optimistic actions may not satisfy the constraint, but they "lead" the played actions toward the optimal action while the played actions stay within the constraint.

For the setting with time-varying cost functions and a fixed constraint (Figure 1b), we consider randomly sampled linear cost functions and a quadratic constraint. In particular, we take the cost functions to be $f_t(x) = \theta_t^\top x$ with $\theta_t \sim \mathcal{U}[0\ 1]^d$ and the constraint function to be of the form $g(x) = a\|x - b\|^2 + c$ where $d = 2$. We consider 10 randomly sampled settings where $a \sim \mathcal{U}[1, 10]$, $b \sim \mathcal{U}[-0.5\ 0.5]^d$ and $c = -a$ in each trial. For each setting, we run ROGD and online gradient descent (OGD), from [1], for each $T \in \{1 \times 10^2, 2 \times 10^2, ..., 5 \times 10^4\}$ with $\eta = \frac{D}{G\sqrt{T}}$, where $D = 2$ and $G = \sqrt{2}$. We give ROGD the information that g is 1-strongly convex and 20-smooth and we give OGD the entire constraint function. For both algorithms, the average static regret (i.e. R_T^s/T) is shown in Figure 1b with the average over all settings shown as a line and ± 1 standard deviation shown as a shaded region. For both ROGD and OGD, the value of R_T^s/T appears to go to zero as T grows, suggesting that the regret is sublinear for this setting.

For the setting with time-varying cost functions and constraints (Figure 1c), we consider a smoothly changing linear cost function and constraint. Specifically, we take the cost functions to be $f_t(x) = \theta_t^\top x$ where $\theta_t^\top$ varies with a constant increment from $[1\ 0]$ to $[-1\ 0]$ along the unit circle and the constraint function to be $g_t(x) = \|x\|^2 + c_t$ where c_t varies with constant increment from -1 to -2. It follows that $P_T \leq \bar{P}_T = 2\pi$. We run ROGD and OGD in this setting for each $T \in \{1 \times 10^2, 2 \times 10^2, ..., 3 \times 10^4\}$ with $\eta = \sqrt{(\bar{P}_T + 1)/T}$. We give ROGD the information that g is 1-strongly convex and 5-smooth and we give OGD the entire constraint function. The average dynamic regret is shown in



(a) Played actions and optimistic actions of ROGD in setting with a fixed cost function.

(b) Average static regret of ROGD and OGD in a setting with time-varying cost functions and a fixed constraint.

(c) Average dynamic regret of ROGD and OGD in a setting with time-varying cost functions and constraints.

Fig. 1: Simulation results of our algorithm ROGD with only first-order feedback of the constraint and the existing algorithm OGD with full knowledge of the constraint in settings with a fixed cost function and fixed constraint (a), time-varying cost functions and fixed constraint (b) and time-varying cost functions and constraints (c).

Figure 1c, which indicates that R_T^d/T goes to 0 as T grows. This suggests that the dynamic regret of both algorithms is sublinear for this setting.

Note that in all of the discussed settings, we use ROGD with a simplified update for γ_t in line 6. In particular, we set $\gamma_t = 1$ if $\tilde{x}_{t+1}$ is in $\mathcal{Y}_t^p$ and $\gamma_t = \frac{1}{\kappa}$ otherwise. The theoretical performance guarantees still hold with this modification and the safety guarantees hold given Lemma 1.

VII. Conclusion

In this work, we study an online convex optimization problem where the player needs to ensure that an unknown constraint is satisfied at all rounds using only first-order feedback at the chosen actions. To address this problem, we propose the algorithm ROGD and prove that it enjoys $\mathcal{O}(\sqrt{T})$ static regret under the assumption that the constraint function is smooth and strongly convex. This algorithm works by using an overestimate of the constraint set to guide the algorithm towards low-regret actions, and using an underestimate of the constraint set to ensure that the played actions satisfy the constraints. We find this approach also works in the more general setting where the constraints are allowed to vary arbitrarily, provided that the constraint sets are monotone and the algorithm receives feedback on the constraint at the next time step. Numerical experiments are given to illustrate our algorithmic approach and validate the theoretical guarantees.

Some interesting directions for future work include (a) investigating if the strong-convexity assumption on the constraint can be relaxed, (b) studying the same problem setting with only zero-order feedback, and (c) seeing if the assumptions made in the time-varying setting (i.e. monotone constraint sets, predictions on next constraint, known path length) can be removed.

References

[1] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proceedings of the 20th international conference on machine learning (icml-03)*, 2003, pp. 928–936.

[2] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, *et al.*, "Ad click prediction: a view from the trenches," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1222–1230.

[3] S. Balseiro, H. Lu, and V. Mirrokni, "Dual mirror descent for online allocation problems," in *International Conference on Machine Learning*. PMLR, 2020, pp. 613–628.

[4] H. Yu and M. J. Neely, "Learning-aided optimization for energy-harvesting devices with outdated state information," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1501–1514, 2019.

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

[6] A. Lesage-Landry, H. Wang, I. Shames, P. Mancarella, and J. A. Taylor, "Online convex optimization of multi-energy building-to-grid ancillary services," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2416–2431, 2019.

[7] S.-J. Kim and G. B. Giannakis, "An online convex optimization approach to real-time energy pricing for demand response," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2784–2793, 2016.

[8] S. Mannor, J. N. Tsitsiklis, and J. Y. Yu, "Online learning with sample path constraints." *Journal of Machine Learning Research*, vol. 10, no. 3, 2009.

[9] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2503–2528, 2012.

[10] H. Guo, X. Liu, H. Wei, and L. Ying, "Online convex optimization with hard constraints: Towards the best of two worlds and beyond," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 426–36 439, 2022.

[11] S. Chaudhary and D. Kalathil, "Safe online convex optimization with unknown linear safety constraints," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6175–6182.

[12] T.-J. Chang, S. Chaudhary, D. Kalathil, and S. Shahrampour, "Dynamic regret analysis of safe distributed online optimization for convex and non-convex problems," *arXiv preprint arXiv:2302.12320*, 2023.

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

[14] H. Yu, M. Neely, and X. Wei, "Online convex optimization with stochastic constraints," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] X. Cao and K. R. Liu, "Online convex optimization with time-varying constraints and bandit feedback," *IEEE Transactions on automatic control*, vol. 64, no. 7, pp. 2665–2680, 2018.
[16] X. Cao, J. Zhang, and H. V. Poor, "A virtual-queue-based algorithm for constrained online convex optimization with applications to data center resource allocation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 703–716, 2018.
[17] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Transactions on Signal Processing*, vol. 68, pp. 731–746, 2020.
[18] M. Castiglioni, A. Celli, A. Marchesi, G. Romano, and N. Gatti, "A unifying framework for online safe optimization," in *NeurIPS ML Safety Workshop*, 2022.
[19] K. Levy and A. Krause, "Projection free online learning over smooth sets," in *The 22nd international conference on artificial intelligence and statistics*. PMLR, 2019, pp. 1458–1466.
[20] Z. Mhammedi, "Efficient projection-free online convex optimization with membership oracle," in *Conference on Learning Theory*. PMLR, 2022, pp. 5314–5390.
[21] D. Garber and E. Hazan, "A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1493–1528, 2016.
[22] E. Hazan and E. Minasyan, "Faster projection-free online learning," in *Conference on Learning Theory*. PMLR, 2020, pp. 1877–1893.
[23] B. Kretzu and D. Garber, "Revisiting projection-free online learning: the strongly convex case," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3592–3600.
[24] H. Yu and M. J. Neely, "A low complexity algorithm with $O(\sqrt{T})$ regret and $O(1)$ constraint violations for online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 21, no. 1, pp. 1–24, 2020. [Online]. Available: http://jmlr.org/papers/v21/16-494.html
[25] M. Bernasconi de Luca, E. Vittori, F. Trovò, and M. Restelli, "Conservative online convex optimization," in *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*. Springer, 2021, pp. 19–34.
[26] A. Moradipari, S. Amani, M. Alizadeh, and C. Thrampoulidis, "Safe linear thompson sampling with side information," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3755–3767, 2021.
[27] A. Pacchiano, M. Ghavamzadeh, P. Bartlett, and H. Jiang, "Stochastic bandits with linear constraints," in *International conference on artificial intelligence and statistics*. PMLR, 2021, pp. 2827–2835.
[28] I. Usmanova, A. Krause, and M. Kamgarpour, "Safe convex learning under uncertain constraints," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2106–2114.
[29] ——, "Safe non-smooth black-box optimization with application to policy search," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 980–989.
[30] B. Guo, Y. Jiang, M. Kamgarpour, and G. Ferrari-Trecate, "Safe zeroth-order convex optimization using quadratic local approximations," in *2023 European Control Conference (ECC)*. IEEE, 2023, pp. 1–8.
[31] D. Garber and E. Hazan, "Faster rates for the frank-wolfe method over strongly-convex sets," in *International Conference on Machine Learning*. PMLR, 2015, pp. 541–549.
[32] E. Hazan and N. Megiddo, "Online learning with prior knowledge," in *Learning Theory: 20th Annual Conference on Learning Theory, COLT 2007, San Diego, CA, USA; June 13-15, 2007. Proceedings 20*. Springer, 2007, pp. 499–513.
[33] D. Anderson, G. Iosifidis, and D. J. Leith, "Lazy lagrangians for optimistic learning with budget constraints," *IEEE/ACM Transactions on Networking*, 2023.

APPENDIX

A. Proof of Theorem 2

In this appendix, we prove Theorem 2, which gives dynamic regret bounds on ROGD in the setting with time-varying constraints. Before getting to the proof of the theorem, we first extend Lemmas 1, 2 and 3 to this setting as follows.

Lemma 4: For ROGD (Algorithm 1) in the setting with time-varying constraints (specified in Section V), we have that $\gamma_t \geq 1/\kappa$ for all $t \in [T]$.

Proof: Since $x_t \in \mathcal{G}_t$ by definition and $\mathcal{G}_t \subseteq \mathcal{G}_{t+1}$, it holds that $g_{t+1}(x_t) \leq 0$. Therefore, the proof of Lemma 1 applies replacing g with g_{t+1}. ∎

Lemma 5: Consider ROGD (Algorithm 1) in the setting with time-varying constraints (specified in Section V). Also, fix any $\epsilon > 0$. If $\eta = \frac{1/\kappa}{G(1-1/\kappa)}\epsilon$, then it holds that $\|x_t - \tilde{x}_t\| \leq \epsilon$ for all t.

Proof: Note that $x_t \in \mathcal{Y}_t$ and $\mathcal{Y}_t \subseteq \mathcal{Y}_{t+1} \subseteq \mathcal{Y}_t^o$ so it follows that $x_t \in \mathcal{Y}_t^o$. Therefore, the proof of Lemma 2 applies. ∎

Lemma 6: Consider ROGD (Algorithm 1) in the setting with time-varying constraints (specified in Section V). Then, for any $v \in \mathcal{Y}_t$, it holds that

$$\nabla f_t(x_t)^\top(\tilde{x}_t - v) \leq \frac{1}{2\eta}\left(\|\tilde{x}_t - v\|^2 - \|\tilde{x}_{t+1} - v\|^2\right) + \frac{1}{2}\eta G^2,$$

for all $t \in [T]$.

Proof: We have that $v \in \mathcal{Y}_t \subseteq \mathcal{Y}_{t+1} \subseteq \mathcal{Y}_t^o$, so we can use the proof of Lemma 3. ∎

With these lemmas established, we prove Theorem 2 in the following.

Proof: Due to the fact that x_t^* is in $\mathcal{Y}_t$ by definition, we can use Lemma 6 with $v \leftarrow x_t^*$ and sum over t to get

$$\begin{aligned}
&\sum_{t=1}^{T} \nabla f_t(x_t)^\top(\tilde{x}_t - x_t^*) \\
&\leq \frac{1}{2\eta}\sum_{t=1}^{T}(\|\tilde{x}_t - x_t^*\|^2 - \|\tilde{x}_{t+1} - x_t^*\|^2) + \frac{1}{2}G^2\eta T \\
&= \frac{1}{2\eta}\sum_{t=1}^{T}(\|\tilde{x}_t\|^2 - \|\tilde{x}_{t+1}\|^2 + 2(\tilde{x}_{t+1} - \tilde{x}_t)^\top x_t^*) \\
&\quad + \frac{1}{2}G^2\eta T \\
&= \frac{1}{2\eta}(\|\tilde{x}_1\|^2 - \|\tilde{x}_{T+1}\|^2 + 2\sum_{t=2}^{T}\tilde{x}_t^\top(x_{t-1}^* - x_t^*) \\
&\quad + 2\tilde{x}_{T+1}^\top x_T^* - 2\tilde{x}_1^\top x_1^*) + \frac{1}{2}G^2\eta T \\
&= \frac{1}{2\eta}(\|\tilde{x}_1 - x_1^*\|^2 - \|\tilde{x}_{T+1} - x_T^*\|^2 + \|x_T^*\|^2 - \|x_1^*\|^2 \\
&\quad + 2\sum_{t=2}^{T}\tilde{x}_t^\top(x_{t-1}^* - x_t^*)) + \frac{1}{2}G^2\eta T \\
&\leq \frac{1}{\eta}(D^2 + D\sum_{t=2}^{T}\|x_{t-1}^* - x_t^*\|) + \frac{1}{2}G^2\eta T \\
&\leq \frac{1}{\eta}D^2 + \frac{1}{\eta}DP_T + \frac{1}{2}G^2\eta T.
\end{aligned}$$

The proof is completed similar to Theorem 1. ∎