Online Convex Optimization With Long-Term Constraints for Predictable Sequences

Deepan Muthirayan, Member, IEEE, Jianjun Yuan, and Pramod P. Khargonekar, Life Fellow, IEEE

Abstract—In this letter, we investigate the framework of Online Convex Optimization (OCO) for online learning. OCO offers a very powerful online learning framework for many applications. In this context, we study a specific framework of OCO called OCO with long-term constraints. Long-term constraints are introduced typically as an alternative to reduce the complexity of projection at every update step in online optimization. While many algorithmic advances have been made towards online optimization with long-term constraints, these algorithms typically assume that the sequence of cost functions over a certain $T$ finite steps that determine the cost to the online learner are adversarially generated. In many circumstances, the sequence of cost functions may not be unrelated, and thus predictable from those observed till a point of time. In this letter, we study the setting where the sequences are predictable. We present a novel algorithm for online optimization with long-term constraints that can leverage such predictability for linear cost functions. We show that, with a predictor that can supply the gradient information of the next function in the sequence, our algorithm can achieve an overall regret and constraint violation rate that is strictly less than the rate that is achievable without prediction.

Index Terms—Online learning, online convex optimization, long-term constraints, prediction.

I. INTRODUCTION

Learning can be key to intelligent decision making especially under circumstances where the underlying environment is uncertain or unknown. Many practical circumstances are scenarios where the learner does not have access to large history of data and has to learn from data gathered online, i.e., while operating in the environment. Such a learning problem is broadly termed as online learning. Online media platforms like Netflix, Spotify etc. [1] leverage online learning techniques in their algorithms. On such platforms, the central problem is to decide what to recommend to the users. Given that the platforms do not have access to prior user data, the platforms will have to learn the users’ preferences online. It is intuitively clear that online learning is relevant to dynamical systems and control. It is noteworthy that the application of online learning to control has received significant interest in the recent years [2], [3]. There are several other practical applications of online learning like portfolio selection, online display advertising [4], learning from experts [1] etc.

In a typical online learning setting, the online learner can make its decision utilizing the history of data it has seen so far. For example, in the context of recommendation systems, it can decide which movies or shows to recommend to a user given the history of user’s response to the recommendations. Since the decisions are made online, the decision typically incurs a cost. In the above example, the cost is a function of user’s response to the recommendations. This is because, typically, the users’ positive response to the recommendations translates into profits. The incurred cost serves as feedback to the online learner, which it can then use to improve its decision in the next iteration. The cost feedback can be broadly classified as: (i) full information feedback and (ii) bandit feedback. Full information feedback is the case where the feedback can be used to infer the full cost function for the time step and bandit feedback is the case where the feedback is just the incurred cost, which cannot be directly used to infer the cost function. The effectiveness of the online learner lies in how effectively it is able to learn with the incremental feedback it receives at every step along the way.

Online Convex Optimization (OCO) is a very widely studied online decision framework for such online learning scenarios. The power and wide reach of OCO lies in the fact that it is applicable to many of the online learning scenarios mentioned above [1]. We study a specific extension of the standard OCO framework, the scenario when the underlying sequence of costs are predictable. The traditional OCO framework assumes that the sequence of cost functions is arbitrary and has no patterns. In several practical scenarios, this can be a very permissive assumption. It is conceivable that the predictability inherent in the sequence of costs can be leveraged to improve the sequence of decisions made online. Not surprisingly, the idea of predictions (assuming the predictions are supplied by an oracle) have been used to improve OCO algorithms (see [5], [6]).

In this letter, we specifically consider the development of OCO algorithms that can leverage predictions for an extension...
of the OCO framework called OCO with long-term constraints. OCO with long-term constraints was introduced as a framework to reduce the computational burden that arises as a result of the projection step at every update step. We elaborate the framework of OCO with long-term constraints and its relevance in the next section. We then review a standard algorithm for OCO with long-term constraints and then introduce our algorithm that can leverage the predictions offered by an oracle.

A. Contribution

We present a novel algorithm for OCO with long-term constraints that can leverage predictions for linear cost functions. We show that, with a predictor that can supply the gradient information with error $O(T^{-d/2})$, $d \in (0,1)$ (which covers all scenarios), our algorithm can achieve a $O(T^{1-d/2})$ regret and a $O(T^{1/4+d/4})$ constraint violation, i.e., an overall rate that is strictly less than $O(\sqrt{T})$, the achievable rate without prediction. Our main contribution is the algorithmic methodology itself. Our proof technique draws on the techniques from the literature in online optimization. Specifically, our analysis technique builds on the most recent approach [7], but can incorporate time-varying constraints unlike [7]. We state that it is an open problem to simultaneously reduce both the regret and constraint violation from what is achievable without prediction.

B. Related Works

The problem of regret minimization with long-term constraints has been extensively studied in [7], [8], [9], [10], [11], [12]. The literature so far has established that a regret of $O(\sqrt{T})$ and a cumulative constraint violation of $T^{1/4}$ can be achieved. But none of these works study the OCO problem with predictability in the cost function sequence.

Many prior works [13], [14] study the problem of optimization with some notion of predictability. In [6], the authors study the case where the prediction may not be helpful at all times. The authors present an algorithm that smoothly interpolates between the two extreme cases, i.e., when the hints are good at all times and when the hints are bad at all times. In [5], the authors present an alternate approach to the problem of optimizing with predictions (or hints). They derive a regret bound in terms of the cumulative error of the predictions (or hints). But none of these works consider the constraints formulated as long-term constraints.

C. Notations

We use $\| \cdot \|$ for the norm. Additionally, $\| \cdot \|_p$ denotes the $p$-norm, for e.g., $\| \cdot \|_2$ denotes the two norm and $\| \cdot \|_1$ denotes the one norm. We denote the transpose of a vector $x$ by $x^\top$. We denote the projection on to a set $\mathcal{K}$ by $\text{Proj}_{\mathcal{K}}(\cdot)$. We denote the sequence $\{x_1, \ldots, x_T\}$ compactly by $x_{1:T}$. We denote the Bregman divergence of a strongly convex function $\mathcal{R}$ by $D_{\mathcal{R}}(\cdot, \cdot)$, $(x, y)$ denotes the standard inner product of two vectors $x, y$. The function $\lfloor \cdot \rfloor_+$ denotes the non-negative component of the input.

II. Problem Formulation

We consider the problem of online optimization with long-term constraints and with predictable sequences. We adopt the standard notation. Every iteration or step in online optimization is typically indexed by time $t$ and is associated with a convex cost function $f_t(\cdot)$. At every step $t$, the decision maker makes a decision, $x_t \in \mathcal{X} \subseteq \mathbb{R}^p$ and incurs the cost $f_t(x_t)$. Typically, the decision has to lie within a set $\mathcal{K} \subseteq \mathcal{X}$. The challenge is that the cost functions are unknown apriori and can be chosen by an adversary. Thus, this problem can be seen as a game between the decision maker and an adversary which can choose the cost functions arbitrarily. The decision maker upon its decision $x_t$ receives a feedback on its decision. The feedback can be full information, in which case the full cost function $f_t(\cdot)$ or the gradient of the cost $\partial f_t(x_t)$ is revealed along with the realized cost $f_t(x_t)$, or the feedback can be bandit, in which case no further information is revealed apart from the realized cost $f_t(x_t)$. Thus, the decision maker can use this information to improve the decision at the next time step. The goal is to compute a sequence of decisions $x_1, x_2, \ldots$ with the feedback received at every step along the way to minimize the regret

$$R_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x),$$

where $\min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)$ is the full cost of the optimal decision [1].

To give an example, in recommendation systems, the optimal decision is the optimal recommendation to all the users and the regret as defined above captures how well the system is able to learn and improve its recommendation over the course of time. Thus, the regret is a well defined measure of how well the decision maker is able to learn online.

For the above problem, it is well known that the simple and projection based online gradient algorithm,

$$x_{t+1} = \text{Proj}_{\mathcal{K}}(x_t - \eta \partial f_t(x_t)),$$

where $\text{Proj}_{\mathcal{K}}(\cdot)$ denotes the projection on to the set $\mathcal{K}$, when $\mathcal{K}$ is closed and convex, achieves $O(\sqrt{T})$ when the sub-gradients are bounded [15], where $O(\sqrt{T})$ has been shown to be tight.

Despite the simplicity of the gradient algorithm, the projection step of the algorithm can be computed easily or in closed form only for simple sets such as a ball or a box. In general, the projection step can be computationally expensive for a general convex set of the form $\mathcal{K} : \{x : g_i(x) \leq 0\}$, where $g(x) = [g_1(x), g_2(x), \ldots, g_m(x)]^\top$, with $g^i : \mathbb{R}^p \rightarrow \mathbb{R}$ being a convex function. An alternate framework was introduced in [8] to simplify the approach. Here, instead of requiring that $g(x) \leq 0$ in each step, the constraint is required to be satisfied only in the long run. Thus, this formulation is popularly referred to as OCO with long-term constraints. Therefore, the goal here is, in addition to minimizing the regret, to minimize the cumulative constraint violation given by

$$C_T = \sum_{t=1}^{T} \sum_{j} (g^i_j(x_t))^+.$$

In particular, the objective is to achieve sub-linear regret and sub-linear cumulative constraint violation. This ensures that, in the long run, the decision satisfies the constraints and the regret
only grows sub-linearly. Since these constraints are typically known apriori, we assume that the function sequence $g_t$s are known.

We make the following standard assumption on the cost function $f_t$ and the constraint function $g_t$.

**Assumption 1:** (i) The set $\mathcal{X}$ is a closed and a convex set. (ii) The constraint function $g_t$ is convex for all $t$. (iii) The cost function $f_t$ is linear, i.e., $f_t(x) = \langle c_t, x \rangle$. (iv) The sub-gradient $\partial f_t(x) = c_t \leq G$, $\partial g_t \leq G \forall x \in \mathcal{X}$, where $G$ is a constant. (v) The functions $f_t(x) \leq F, g_t(x) \leq F, \forall x \in \mathcal{X}$. (vi) The decision maker receives full information feedback. (vii) There is a feasible $x$ such that $g_t(x) \leq 0$ for all $t$.

The assumption that the cost functions are linear is standard in the OCO setting with predictable sequences; see [5], [6]. The other assumptions are also standard in the online optimization literature [7]. As a start, we focus here on the full information feedback setting and we believe our algorithms can be extended to the bandit feedback setting just as in the setting without prediction.

As motivated earlier, the assumption that the sequence of cost functions are arbitrary can be harsher. The idea here is that the sequence of costs might not be totally arbitrary and adversarial but have a relation. In such circumstances, it is possible that the decision maker can predict the next cost from the past information, and thus improve its performance. Therefore, we consider the setting where the sequence of cost functions are predictable and there is an oracle that supplies a prediction $h_t$ of the gradient $c_t$ before time $t$, very much in the spirit of [5], [6]. Our goal is to develop an algorithm for the OCO setting with long-term constraints that can leverage such predictions to improve the regret while keeping the constraint violation controlled.

### III. Review: OCO Algorithm

Here, we review the most recent approach to online optimization with long-term constraints formulation [7]. We present the algorithm and its properties for the case where the constraint functions are time-invariant, i.e., $g_t = g$. A standard algorithm for OCO with long-term constraints is shown in Algorithm 1.

#### Algorithm 1 Online Optimization Algorithm Without Prediction [7, Algorithm 1]

1: **Initialize:** $x_1 \in \mathcal{X}, q_0 = 0, c \in (0, 1), \gamma_1 = T^{c/2}/(\sqrt{2}G)$
2: for $t = 1, \ldots, T$ do
3:   Apply $x_t$; suffer cost $f_t(x_t)$; suffer constraint violation $[g_t(x_t)]_+$
4:   Observe $c_t$
5:   Set $\eta_t = 1/T^c$ and $\gamma_{t+1} = T^{c/2}/(\sqrt{2}G)$
6:   Update
   \[q_t = q_{t-1} + \gamma_t g_t(x_t)]_+, \hat{q}_t = q_t + \gamma_t g_t(x_t)]_+,\]
   \[x_{t+1} = \arg\min_{x \in \mathcal{X}} \eta_t(x, c_t)\]
   \[+ \eta_t \gamma_{t+1} \hat{q}_t, [g_t(x)]_+] + \|x - x_t\|_2^2.\]
7: end for

#### Algorithm 2 Online Optimization Algorithm With Prediction

1: **Input:** $R$—strongly convex function w.r.t $\|\cdot\|_2$
2: **Initialize:** $x_1 = z_1 = \arg\min_{z \in \mathcal{X}} R(z), q_0 = 0, c \in (0, 1)$
3: for $t = 1, \ldots, T$ do
4:   Apply $x_t$; suffer cost $f_t(x_t)$; suffer constraint violation $[g_t(x_t)]_+$
5:   Observe $c_t$; observe the hint $h_{t+1}$
6:   Set $\eta_{t+1} = 1/T^c$ and $\gamma_{t+1} = T^{c/2}/G$
7:   Update
   \[q_t = q_{t-1} + \gamma_t [g_t(x_t)]_+,\]
   \[z_{t+1} = \arg\min_{z \in \mathcal{X}} \eta_{t+1} (z, c_t)\]
   \[+ \eta_{t+1} \gamma_{t+1} \hat{q}_t, [g_t(x)]_+] + D_R(z, z_t).\]
8: Update
   \[\hat{q}_t = q_t + \gamma_{t+1} [g_t(z_{t+1})]_+,\]
   \[x_{t+1} = \arg\min_{x \in \mathcal{X}} \eta_{t+1} (x, h_{t+1})\]
   \[+ \eta_{t+1} \gamma_{t+1} \hat{q}_t, [g_t(x)]_+] + D_R(x, z_{t+1}).\]
9: end for

The key difference in this algorithm compared to a standard OCO algorithm is the optimization step carried out to compute $x_{t+1}$. A standard online algorithm just updates the decision along the direction of the gradient of the current cost with an appropriate step size. In this case though, the decision is updated along a direction that is a combination of the gradient of the cost function and a direction that minimizes the cumulative constraint violation. This additional direction compared to the standard algorithm is contributed by the second term inside the optimization carried out to update $x_t$ to $x_{t+1}$. This design achieves the desired properties in the long run, i.e., achieves sub-linear constraint violation and sub-linear regret. We summarize the properties of the algorithm below.

**Theorem I** [7, Th. 1]: Suppose Assumption 1 holds. Suppose $\eta_t = 1/T^c$, $\gamma_t = \frac{1}{\sqrt{T}^c}$, where $c \in (0, 1)$. Then, Algorithm 1 achieves
\[R_T = O(T^{max\{1-c, c\}}, C_T \leq T^{1/2-c/2}.\]
The proof of this theorem is available in [7].

### IV. OCO Algorithm With Prediction

In this scenario, the following information is available to decide $x_t$ at $t$: the prediction $h_t$ for the gradient of the cost function at $t$, in addition to the gradient of the cost functions at all previous time steps. With prediction, the algorithm can afford to anticipate and therefore make better decisions. The online algorithm we propose leverages the prediction $h_t$ to make its decision $x_t$ at time $t$. The complete algorithm is outlined in Algorithm 2.

Our algorithm has two update steps. In each update step a constraint violation cost is included, which is needed to simultaneously minimize the constraint violation alongside the regret. The intuition for this inclusion is the same as Algorithm 1. The two-step decision update is similar to the
two-step update in the standard online optimization with prediction [5]. The first step computes a decision like in the regular online optimization with long-term constraints. The second update step updates the decision further along the prediction $h_t$ as in the standard online update with prediction. The difference here is that the optimization objective for this step like in the first step includes an additional constraint violation cost. The additional constraint violation cost included in the second update step has the same form as the constraint violation cost in standard online update with long-term constraints [7]. The major difference compared to the standard Algorithm 1 is the two-step nature of the update itself. We note that there are also differences in how $\hat{q}_t$s are defined and the constraint violation cost in each update step. In addition, the constraints have to be time-invariant for Algorithm 1 to work, whereas they can be time-varying in our case.

Next, we present the regret and the cumulative constraint violation our algorithm can achieve with prediction.

**Theorem 2:** Suppose Assumption 1 holds. Suppose $\eta_t = \eta = 1/T^c$, $y_t = y = 1/\sqrt{T} \gamma$, where $c \in (0, 1)$. Then, under Algorithm 2, when $\|c_t - h_t\| = O(T^{-a/2})$, 
$$R_T \leq O(T^{\max\{1-a-c, c\}}), \quad C_T \leq T^{1/2-c/2}.$$ 

**Discussion:** First, we note that, we recover the state-of-the-art result when the prediction does not provide any valuable information, that is, when $\eta = 0$: $O(\sqrt{T})$ for regret and $O(T^{1/4})$ for cumulative constraint violation. We can recover the same result for any $a$ by setting $c = 1/2$. But, we can do better and reduce the regret further. When the prediction is the worst, since $c_t \leq G$, $\|c_t - h_t\| = O(G) = O(1)$. We can set $a$ tighter when the prediction error is small. For example, when $\|c_t - h_t\| = O(T^{-b})$, where $b \in (0, 1/2)$, we can set $a = 2b$, and when $\|c_t - h_t\| = O(T^{-b})$, where $b \geq 1/2$, we can still set $a < 1$. Across these scenarios, setting $c = (1 - a)/2$ gives us $O(T^{1-a/2})$ regret and $O(T^{1/4+a/4})$ constraint violation, which gives us a much better regret but with an increase in constraint violation. The upside is that the constraint violation rate is still within $o(\sqrt{T})$, which is strictly less than the $O(\sqrt{T})$ overall rate that is achievable in any form OCO without prediction. Here, the designer can choose to set $a$, for e.g., when $b \geq 1/2$, depending on how much less of constraint violation, given by $O(T^{1/4+a/4})$, is desirable compared to $O(\sqrt{T})$. We state that it is an open problem to simultaneously reduce regret and constraint violation. We suspect this could be a result of certain theoretical bottlenecks prevalent in all OCO analysis. We point to this bottleneck in the next section.

**V. MAIN RESULTS**

We break the analysis into two main steps. In the first lemma, we derive two key inequalities relating the variables of the problem. We then use this intermediate lemma to prove our theorem.

**Lemma 1:** Suppose Assumption 1 holds. Then, for any $x^* \in X$, under Algorithm 2

$$\sum_{t=1}^{T} y_t(q_{t-1}, [g_t(x_t)])_+ + R_T \leq \sum_{t=1}^{T} \eta_t \|c_t - h_t\|^2 + \sum_{t=1}^{T} \frac{1}{\eta_t} (D_R(z_t, x^*) - D_R(x^*, z_t+1)) - \sum_{t=1}^{T} \frac{1}{\eta_t} D_R(x_t, z_t)$$

$$+ \sum_{t=1}^{T} y_t(q_{t-1}, [g_t(x_t)])_+ - \sum_{t=1}^{T} \sum_{j=1}^{m} y_t^j \|g_t(z_t)_+ - [g_t(x_t)]_+\|,\quad \frac{1}{2} \|q_T\|^2 \leq \sum_{t=1}^{T} \eta_t \|c_t - h_t\|^2 + \sum_{t=1}^{T} y_t(q_{t-1}, [g_t(x_t)])_+ - R_T$$

$$+ \sum_{t=1}^{T} \frac{1}{\eta_t} (D_R(z_t, x^*) - D_R(x^*, z_t+1)) + \sum_{t=1}^{T} (y_t^2 G_t^2 - \frac{1}{\eta_t}) D_R(x_t, z_t).$$

Please see Appendix A for the proof. In the next theorem, we use the above lemma to derive a bound on the regret and cumulative constraint violation in terms of the cumulative error in the prediction.

**Theorem 3:** Suppose Assumption 1 holds, $\eta_t = \eta$, $y_t = y = 1/\sqrt{T} \gamma$, then for any $x^* \in \{x : g_t(x) \leq 0, \forall t\}$, Algorithm 2 achieves

$$R_T \leq \sum_{t=1}^{T} \frac{\eta}{2} \|c_t - h_t\|^2 + \frac{1}{\eta} D_R(z_1, x^*), \quad C_T \leq \frac{1}{\gamma} \sum_{t=1}^{T} \|c_t - h_t\|^2 + \frac{1}{\eta} D_R(z_1, x^*) + FT.$$ 

Please see Appendix B for the proof. It is easy to see that when the prediction error is not very small, i.e., $\|c_t - h_t\| = O(1)$, the regret upper bound reduces to the standard bound of the form $O(\sqrt{T}) + O(1/\eta)$. Here, we can set the step rate $\eta = O(T^{1/2})$ and recover the rate that is achievable without prediction. Similarly, for this case, we can recover the rate $O(T^{1/4})$ for cumulative constraint violation, and thereby recover the scaling that is achievable without prediction.

When the prediction error is small, lets consider the case where $\|c_t - h_t\| \leq 1/\sqrt{T} \gamma$. With $\eta_t$ set as $\eta_t = \eta = T^{-c}$, where $c$ is some constant and $c \in (0, 1)$, it follows that

$$\sum_{t=1}^{T} \|c_t - h_t\|^2 \leq O(T^{1-a-c}), \quad \frac{1}{\eta} D_R(z_1, x^*) \leq O(T^c)$$

$$\Rightarrow R_T \leq O(T^{\max\{1-a-c, c\}}).$$

Similarly, we have that

$$\sum_{t=1}^{T} \|c_t - h_t\|^2 + \frac{1}{\eta} D_R(z_1, x^*) \leq O(T^{\max\{1-a-c, c\}})$$

$$\Rightarrow C_T \leq O(T^{\max\{1-a-c, 1\} / 2 + 1/2 - c/2}).$$

This completes the analysis and proof of Theorem 2. The bottleneck arises from “1” of the terms input to the max function: max\{1 – a – c, c, 1\}. This arises from the $FT$ term in the upper bound to $C_T$, which ends up dominating the other two terms. The term $FT$ arises from having to bound $-R_T$. To the best of our knowledge, there are no known better bounds than $FT$ and this is the technical bottleneck.
VI. CONCLUSION

In this letter, we consider the development of an algorithm that can leverage the predictability in the sequence of cost functions in online optimization. We specifically consider the online optimization setting with the constraints formulated as long term constraints. For this setting, we present a novel online optimization algorithm that can leverage predictions. We show that our algorithm can achieve an overall regret and cumulative constraint rate that is strictly less than $O(\sqrt{T})$, the rate that is achievable without prediction. We also present some open challenges.

APPENDIX A

PROOF OF Lemma 1

For any $x^* \in X$,

$$f_t(x_t) - f_t(x^*) = \langle x_t - x^*, c_t \rangle = \langle x_t - z_{t-1}, c_t - h_t \rangle + \langle x_t - z_{t-1}, h_t + \gamma_t \hat{g}_t(x_t) + \hat{q}_{t-1} \rangle$$

By A.1, we get

$$\langle x_t - z_{t-1}, c_t - h_t \rangle \leq \|x_t - z_{t-1}\| \|c_t - h_t\|\leq \frac{1}{2\eta_t} \|x_t - z_{t-1}\|^2 + \frac{\eta_t}{2} \|c_t - h_t\|^2.$$  

(2)

Now, for any convex function $h(\cdot)$, any update of the form $a^* = \arg\min_{a \in A} h(a) + D_R(a, b)$ (see [7]), for any $d \in A$, satisfies

$$\langle a^* - d, \partial h(a^*) \rangle \leq D_R(d, b) - D_R(d, a^*) - D_R(a^*, b).$$

Applying this to the second term with $h(a) = \langle a, h_t \rangle + \gamma_t \langle \hat{q}_{t-1}, [g_t(a)]_+ \rangle$, and using the second update step of Algorithm 2, we get

$$\langle x_t - z_{t-1}, h_t + \gamma_t \hat{g}_t(x_t) + \hat{q}_{t-1} \rangle \leq \frac{1}{\eta_t} (D_R(z_t, z_{t-1}) - D_R(z_{t-1}, x_t)).$$

(3)

Applying the same formula to the fourth term with $h(a) = \langle a, c_t \rangle + \gamma_t \hat{q}_{t-1}, \hat{g}_t(z_{t+1})_+ + \hat{q}_{t-1}, a \rangle$, and using the first update step of Algorithm 2, we get

$$\langle z_{t+1} - x^*, c_t + \gamma_t \hat{g}_t(z_{t+1}) + \hat{q}_{t-1} \rangle \leq \frac{1}{\eta_t} (D_R(z_{t+1}, x^*) - D_R(z_{t+1}, z_t)).$$

(4)

Then, using Equations (2), (3), and (4), we get

$$\langle x_t - x^*, c_t \rangle \leq \frac{1}{2\eta_t} \|x_t - z_{t-1}\|^2 + \frac{\eta_t}{2} \|c_t - h_t\|^2 + \frac{1}{\eta_t} (D_R(z_t, z_{t-1}) - D_R(z_{t-1}, x_t))$$

$$+ \frac{1}{\eta_t} (D_R(z_t, x^*) - D_R(x^*, z_{t+1}) - D_R(z_{t+1}, z_t))$$

$$+ \langle z_{t+1} - x^*, \gamma_t [g_t(x_t)]_+ + \hat{q}_{t-1} \rangle$$

$$+ \langle z_{t+1} - x_t, \gamma_t [g_t(x_t)]_+ + \hat{q}_{t-1} \rangle$$

$$+ \langle x^* - z_{t-1}, \gamma_t [g_t(z_{t+1})]_+ + \hat{q}_{t-1} \rangle.$$
\[\begin{align*}
\|g_t(x_t)\|_1 & = \frac{1}{\gamma_t} \left( \|q_t\|_1 - \|q_{t-1}\|_1 \right) \\
& = \frac{1}{\gamma_t} \|q_t\|_1 - \frac{1}{\gamma_{t-1}} \|q_{t-1}\|_1 + \left( \frac{1}{\gamma_{t-1}} - \frac{1}{\gamma_t} \right) \|q_{t-1}\|_1.
\end{align*}\]
Summing over \( t \in \{1, \ldots, T\} \), we get
\[\begin{align*}
\sum_{t=1}^T \|g_t(x_t)\|_1 &= \frac{1}{\gamma_T} \|q_T\|_1 + \sum_{t=1}^{T-1} \left( \frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} \right) \|q_t\|_1.
\end{align*}\]
(7)

Also,
\[\begin{align*}
\|q_t\|^2 &= \|q_{t-1}\| + \gamma_t \|g_t(x_t)\|_1^2 \\
& = \|q_{t-1}\|^2 + 2\gamma_t \|q_{t-1}g_t(x_t)\|_1 + \gamma_t \|g_t(x_t)\|_1^2.
\end{align*}\]

This implies that
\[\begin{align*}
\frac{1}{2} \left( \|q_t\|^2 - \|q_{t-1}\|^2 \right) &= \gamma_t \|q_{t-1}\|_1 \|g_t(x_t)\|_1 + \frac{1}{2} \gamma_t \|g_t(x_t)\|_1^2.
\end{align*}\]
(8)

Next, we have that
\[\begin{align*}
\gamma_t^2 \langle [g_t(z_t)]_+, [g_t(x_t)]_+ \rangle &= \frac{1}{2} \left( \|\gamma_t [g_t(z_t)]_+\|^2 + \|\gamma_t [g_t(x_t)]_+\|^2 \right) \\
& - \gamma_t^2 \|\gamma_t [g_t(z_t)]_+ - [g_t(x_t)]_+\|^2.
\end{align*}\]
Then, using the fact that \( g_t(z_t) - g_t(x_t) \leq G\|z_t - x_t\|_1 \), we get
\[\begin{align*}
\gamma_t^2 \langle [g_t(z_t)]_+, [g_t(x_t)]_+ \rangle &\geq \frac{\gamma_t^2}{2} \left( \|g_t(z_t)\|_1^2 + \|g_t(x_t)\|_1^2 \right) \\
& - \gamma_t^2 G^2 \frac{\gamma_t^2}{2} \|z_t - x_t\|_1^2.
\end{align*}\]
(9)

Then, using Equations (6), (7), (8), and (9), we get
\[\begin{align*}
\frac{1}{2} \left( \|q_t\|^2 - \|q_{t-1}\|^2 \right) + \langle x_t - x^*, c_t \rangle &\leq \frac{\eta_t}{\gamma_t} \|c_t - h_t\|^2 \\
& + \frac{1}{\gamma_t} \left( D_{R}^2(z_t, x^*) - D_{R}^2(x^*, z_{t+1}) \right) \\
& + \left( \gamma_t^2 G^2 - \frac{1}{\eta_t} \right) D_{R}^2(x_t, z_t) \\
& + \sum_{t=1}^T \gamma_t \langle \hat{q}_{t-1}, [g_t(x^*)]_+ \rangle.
\end{align*}\]
Since \( q_0 = 0 \), summing over \( t \in \{1, \ldots, T\} \), we get
\[\begin{align*}
\frac{1}{2} \|q_T\|^2 &\leq \sum_{t=1}^T \frac{\eta_t}{\gamma_t} \|c_t - h_t\|^2 - R_T \\
& + \sum_{t=1}^T \frac{1}{\gamma_t} \left( D_{R}^2(z_t, x^*) - D_{R}^2(x^*, z_{t+1}) \right) \\
& + \sum_{t=1}^T \gamma_t \langle \hat{q}_{t-1}, [g_t(x^*)]_+ \rangle.
\end{align*}\]
This completes the proof.

APPENDIX B

PROOF OF THEOREM 3

From Lemma 1 and since \( x^* \) satisfies \( g_t(x) \leq 0 \) for all \( t \), we have that
\[\begin{align*}
R_T &\leq \sum_{t=1}^T \frac{\eta_t}{\gamma_t} \|c_t - h_t\|^2 \\
& + \sum_{t=1}^T \frac{1}{\gamma_t} \left( D_{R}^2(z_t, x^*) - D_{R}^2(x^*, z_{t+1}) \right).
\end{align*}\]
In the above, the term \( \langle q_{t-1}, [g_t(x_t)]_+ \rangle \) can be ignored because the dot-product is always non-negative as it is the dot-product of two vectors with non-negative elements. Therefore, since \( \eta_t = \eta \) is a constant, we have that
\[\begin{align*}
R_T &\leq \frac{\eta}{\gamma_T} \sum_{t=1}^T \|c_t - h_t\|^2 + \frac{1}{\eta} D_{R}^2(z_1, x^*).
\end{align*}\]
This completes the first part of the proof. Once again, from Lemma 1 and since \( \gamma_t^2 G^2 - \frac{1}{\eta_t} = 0 \) and \( \eta_t = \eta \), we have that
\[\begin{align*}
\frac{1}{2} \|q_T\|^2 &\leq \frac{\eta}{\gamma_T} \sum_{t=1}^T \|c_t - h_t\|^2 + \frac{1}{\eta} D_{R}^2(z_1, x^*) - R_T.
\end{align*}\]
Since \( -R_T \leq FT \), we get that
\[\begin{align*}
\frac{1}{2} \|q_T\|^2 &\leq \frac{\eta}{\gamma_T} \sum_{t=1}^T \|c_t - h_t\|^2 + \frac{1}{\eta} D_{R}^2(z_1, x^*) + FT.
\end{align*}\]
Since
\[\begin{align*}
\sum_{t=1}^T \|g_t(x_t)\|_1 &= \frac{1}{\gamma_T} \|q_T\|_1 + \sum_{t=1}^{T-1} \left( \frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}} \right) \|q_t\|_1,
\end{align*}\]
and \( \gamma_T = \gamma \), we get that
\[\begin{align*}
\sum_{t=1}^T \|g_t(x_t)\|_1 &\leq \frac{1}{\gamma} \sum_{t=1}^T \|c_t - h_t\|^2 + \frac{1}{\eta} D_{R}^2(z_1, x^*) + FT.
\end{align*}\]

REFERENCES

[1] E. Hazan et al., “Introduction to online convex optimization,” Found. Trends Optim., vol. 2, nos. 3–4, pp. 157–325, 2016.
[2] E. Hazan, S. Kakade, and K. Singh, “The nonstochastic control problem,” in Proc. Mach. Learn. Res., 2020, pp. 408–421.
[3] D. Mahdavi, J. Yu, D. Kalathil, and P. P. Khargonekar, “Online learning for receding horizon control with provable regret guarantees,” 2021, arXiv:2111.15041.
[4] A. Goldfarb and C. Tucker, “Online display advertising: Targeting and obtrusiveness,” Market. Sci., vol. 30, no. 3, pp. 389–404, 2011.
[5] A. Rakhlin and K. Sridharan, “Online learning with predictable sequences,” in Proc. Mach. Learn. Res., 2013, pp. 993–1019.
[6] A. Bhaskara, A. Czumosy, R. Kumar, and M. Purohit, “Online learning with imperfect hints,” in Proc. Mach. Learn. Res., 2020, pp. 822–831.
[7] X. Yu, X. Li, T. Yang, L. Xie, T. Chai, and K. H. Johansson, “Regret and cumulative constraint violation analysis for online convex optimization with long term constraints,” 2021, arXiv:2106.05135.
[8] M. Mahdavi, R. Jin, and T. Yang, “Trading regret for efficiency: Online convex optimization with long term constraints,” J. Mach. Learn. Res., vol. 13, no. 1, pp. 2503–2528, 2012.
[9] R. Jenatton, J. Huang, and C. Archambeau, “Adaptive algorithms for online convex optimization with long-term constraints,” in Proc. Mach. Learn. Res., 2016, pp. 402–411.
[10] H. Yu, M. J. Neely, and X. Wei, “Online convex optimization with stochastic constraints,” 2017, arXiv:1708.03741.
[11] J. Yuan and A. Lamperski, “Online convex optimization for cumulative constraints,” 2018, arXiv:1802.06472.
[12] 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,” 2016, arXiv:1604.02218.
[13] E. Hazan and N. Megiddo, “Online learning with prior knowledge,” in Proc. Mach. Learn. Res., 2007, pp. 499–513.
[14] O. Dekel, A. Flajolet, N. Haggatalab, and P. Jaillet, “Online learning with a hint,” in Proc. Mach. Learn. Res., 2017, pp. 5299–5308.
[15] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” in Proc. Mach. Learn. Res., 2003, pp. 928–936.