Tight Bounds for Online Convex Optimization with Adversarial Constraints

Abhishek Sinha [1] **Rahul Vaze** [1]

Abstract

A well-studied generalization of the standard online convex optimization (OCO) is constrained online convex optimization (COCO). In COCO, on every round, a convex cost function and a convex constraint function are revealed to the learner after the action for that round is chosen. The objective is to design an online policy that simultaneously achieves a small regret while ensuring small cumulative constraint violation (CCV) against an adaptive adversary. A long-standing open question in COCO is whether an online policy can simultaneously achieve $O(\sqrt{T})$ regret and $O(\sqrt{T})$ CCV without any restrictive assumptions. For the first time, we answer this in the affirmative and show that an online policy can simultaneously achieve $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV. We establish this result by effectively combining the adaptive regret bound of the AdaGrad algorithm with Lyapunov optimization - a classic tool from control theory. Surprisingly, the analysis is short and elegant.

1. Introduction

We consider the Constrained Online Convex Optimization problem (COCO for short) with adversarially chosen convex cost and convex constraint functions. In COCO, on every round, a convex cost function and a convex constraint function are revealed to the learner, after the action for that round is chosen. The objective is to design an online policy that simultaneously achieves a small regret (w.r.t. cost functions) while ensuring a small cumulative constraint violations (CCV) (w.r.t. constraint functions) against any adaptive adversary that is allowed to choose a single action across all rounds.

The COCO problem has been extensively studied in the literature over the past decade and the best-known bounds for simultaneouls regret and CCV are $O(\sqrt{T})$ and $O(T^{3/4})$, respectively (Guo et al., 2022; Yi et al., 2023; Sinha & Vaze, 2023), without any additional assumptions. A simultaneous lower bound on regret and CCV is known to be $\Omega(\sqrt{T})$ and $\Omega(\sqrt{T})$ (Sinha & Vaze, 2023). Please refer to Table 1 in the Appendix for a summary of the major results.

It is known that with additional assumptions, simultaneous $O(\sqrt{T})$ regret and $O(\sqrt{T})$ CCV is achievable (See the Related Work Section below). However, even after more than a decade of sustained work on this problem, the question: whether an online policy can simultaneously achieve $O(\sqrt{T})$ regret and $O(\sqrt{T})$ CCV, without any additional assumptions remains unresolved and constitutes a major open problem in this area.

In this paper, we resolve this problem for the first time and make the following contributions.

1. We propose an efficient policy that simultaneously achieves $O(\sqrt{T})$ regret and $O(\sqrt{T}\log T)$ CCV for the COCO problem. Our policy breaks the long-standing $O(T^{3/4})$ barrier for the CCV and achieves the lower bound up to a logarithmic term.

2. The proposed policy simply runs the standard AdaGrad algorithm on a specially constructed sequence of surrogate cost functions and requires no explicit parameter tuning. On every round, the policy needs to compute the gradients of the cost and constraint functions at the current action and performs an Euclidean projection.

3. Unlike the primal-dual-based algorithms in the previous works, the design of our policy employs the Lyapunov method from the classical control theory. The presented analysis is short and elegant, may be of independent interest for other related problems.

2. Related Work

While the regret bound of $O(\sqrt{T})$ for COCO is minimax optimal, a number of papers have obtained tighter CCV bounds under various assumptions, such as time-invariant constraints (Mahdavi et al., 2012; Jenatton et al., 2016; Yuan & Lamperski, 2018; Yu & Neely, 2020; Yi et al., 2021), stochastic constraints (Yu et al., 2017; Sinha, 2024), Slater's

[1]School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai 400005, India. Correspondence to: Abhishek Sinha <abhishek.sinha@tifr.res.in>.

Workshop on Foundations of Reinforcement Learning and Control at the 41st International Conference on Machine Learning, Vienna, Austria. Copyright 2024 by the author(s).

condition (Neely & Yu, 2017; Yi et al., 2023), or the non-negativity of the regret (Sinha & Vaze, 2023). Furthermore, the algorithms proposed in Guo et al. (2022); Yi et al. (2023; 2021); Yu & Neely (2020) are complex as they need to solve a full-fledged convex optimization problem on each round.

3. Problem formulation

Consider a repeated game between an online policy and an adaptive adversary. In this game, on each round $t \geq 1$, the policy chooses a feasible action x_t from an admissible set $\mathcal{X}$. The set $\mathcal{X}$ is assumed to be non-empty, closed, and convex with a finite Euclidean diameter D. Upon observing the current action x_t, the adversary chooses two convex functions - a *cost* function $f_t : \mathcal{X} \to \mathbb{R}$ and a *constraint* function $g_t : \mathcal{X} \to \mathbb{R}$. The constraint function g_t correspond to an online constraint of the form $g_t(x) \leq 0$. Since the constraints are adaptively chosen, they cannot be expected to be satisfied by the policy on each round. Consequently, the policy incurs a cost of $f_t(x_t)$ and an instantaneous constraint violation of $\max(0, g_t(x_t))$. Our high-level objective is to design an online policy that achieves a small cumulative cost while approximately satisfying the constraints in the long term. Throughout the paper, we assume that all cost and constraint functions are convex and G-Lipschitz. The regret of any policy is computed by comparing its cumulative cost against that of a fixed feasible action $x^\star \in \mathcal{X}$ that satisfies all constraints on each rounds. Specifically, let $\mathcal{X}^\star \subseteq \mathcal{X}$ be the set consisting of all actions satisfying all constraints:

$$\mathcal{X}^\star = \{x \in \mathcal{X} : g_t(x) \leq 0, \ \forall t \geq 1\}. \tag{1}$$

We assume that the feasible set is non-empty, *i.e.,* $\mathcal{X}^\star \neq \emptyset$. We then define the Regret and the CCV of any policy as

$$\text{Regret}_T = \sup_{x^\star \in \mathcal{X}^\star} \sum_{t=1}^{T} \big(f_t(x_t) - f_t(x^\star)\big), \tag{2}$$

$$\text{CCV}_T = \sum_{t=1}^{T} (g_t(x_t))^+, \tag{3}$$

where we have defined $(y)^+ \equiv \max(0, y), \ y \in \mathbb{R}$.

In the standard OCO problem, only the cost functions are revealed on every round and there is no online constraint function. The goal for the standard OCO problem is to only minimize the regret. Hence, the standard OCO problem can be seen to be a special case of COCO where $g_t = 0, \forall t$, and hence, $\mathcal{X}^\star = \mathcal{X}$.

Note: We may consider an extension of the above problem where $k > 1$ constraints of the form $g_{t,i}(x) \leq 0, i \in [k]$, are revealed on each round. These multiple constraints can be combined to a single constraint by replacing them with a new constraint $g_t(x) \leq 0$, where the function g_t is taken to be the point wise max, *i.e.,* $g_t(x) \equiv \max_{i=1}^{k} g_{t,i}(x)$.

4. Optimal Regret and Constraint Violation Bounds for COCO

4.1. Overview of the technique

Compared to the standard online convex optimization problem where the objective is to minimize the Regret (Hazan, 2022), in this problem our objective is two fold - to *simultaneously* control the Regret and the CCV. Towards this end, in the following, we propose a Lyapunov-based technique that yields $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV.

4.2. Preliminaries

We now briefly recall the first-order methods (*a.k.a.* Projected Online Gradient Descent (OGD)) for the standard OCO problem. These methods differ among each other in the way the step sizes are chosen. For a sequence of convex cost functions $\{\hat{f}_t\}_{t\geq 1}$, a projected OGD algorithm selects the successive actions as (Orabona, 2019, Algorithm 2.1):

$$x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t), \ \ \forall t \geq 1, \tag{4}$$

where $\nabla_t \equiv \nabla \hat{f}_t(x_t)$ is a subgradient of the function $\hat{f}_t$ at x_t, $\mathcal{P}_{\mathcal{X}}(\cdot)$ is the Euclidean projection operator on the set $\mathcal{X}$ and $\{\eta_t\}_{t\geq 1}$ is a specified step size sequence. The (diagonal version of the) AdaGrad policy adaptively chooses the step size sequence as a function of the previous subgradients as $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^{t} G_\tau^2}}$, where $G_t = ||\nabla_t||_2, t \geq 1$ (Duchi et al., 2011). [1] It enjoys the following adaptive regret bound.

Theorem 4.1. *(Orabona, 2019, Theorem 4.14) The AdaGrad policy, with the above step size sequence, achieves the following regret bound for the standard OCO problem:*

$$\textit{Regret}_T \leq \sqrt{2}D\sqrt{\sum_{t=1}^{T} G_t^2}. \tag{5}$$

4.3. Design and Analysis of the Algorithm

To simplify the analysis, we pre-process the cost and constraint functions as follows.

Pre-processing: On every round, we clip the negative part of the constraint to zero by passing it through the standard ReLU unit. Next, we scale both the cost and constraint functions by a factor of $\alpha \equiv (2GD)^{-1}$. More precisely, we define $\tilde{f}_t \leftarrow \alpha f_t, \tilde{g}_t \leftarrow \alpha (g_t)^+$. Hence, the pre-processed functions are $\alpha G = (2D)^{-1}$-Lipschitz with $\tilde{g}_t \geq 0, \forall t$.

In the following, we will derive the Regret and CCV bounds for the pre-processed functions. The corresponding bounds for the original unscaled functions are obtained upon scaling the results back by α^{-1} in the final step.

[1] We set $\eta_t = 0$ if $G_t = 0$.

Algorithm 1 Online Policy for COCO

Input: Horizon length T, Sequence of convex cost functions $\{f_t\}_{t=1}^T$ and constraint functions $\{g_t\}_{t=1}^T$, an upper bound G to the norm of their (sub)-gradients, Diameter D of the admissible set $\mathcal{X}$
Parameters: $V = 1, \lambda = 1/(2\sqrt{T}), \Phi(x) = \exp(\lambda x) - 1, \alpha = (2GD)^{-1}$.
Initialization: Set $x_1 = \mathcal{P}_{\mathcal{X}}(\mathbf{0}), Q(0) = 0$.
for $t = 1 : T$ **do**
  Choose x_t, observe f_t, g_t, incur a cost of $f_t(x_t)$ and constraint violation of $(g_t(x_t))^+$
  $\tilde{f}_t \leftarrow \alpha f_t, \tilde{g}_t \leftarrow \alpha \max(0, g_t)$.
  $Q(t) = Q(t-1) + \tilde{g}_t(x_t)$.
  Compute $\nabla_t = \nabla \hat{f}_t(x_t)$, where $\hat{f}_t$ is defined in (8)
  $x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t)$, where $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^t ||\nabla_\tau||_2^2}}$.
end for

4.3.1. Surrogate cost functions

Let $Q(t)$ denote the CCV for the pre-processed constraints up to round t. Clearly, $Q(t)$ satisfies the simple recursion $Q(t) = Q(t-1) + \tilde{g}_t(x_t), t \geq 1$, with $Q(0) = 0$. Recall that one of our objectives is to make $Q(t)$ as small as possible. Towards this, let $\Phi : \mathbb{R}_+ \to \mathbb{R}_+$ be any non-decreasing convex potential (Lyapunov) function such that $\Phi(0) = 0$. Using the convexity of the function $\Phi(\cdot)$, we have

$$\begin{aligned} \Phi(Q(t)) &\leq \Phi(Q(t-1)) + \Phi'(Q(t))(Q(t) - Q(t-1)) \\ &= \Phi(Q(t-1)) + \Phi'(Q(t))\tilde{g}_t(x_t). \end{aligned} \tag{6}$$

Hence, the one-step change (*drift*) of the potential function $\Phi(Q(t))$ can be upper bounded as

$$\Phi(Q(t)) - \Phi(Q(t-1)) \leq \Phi'(Q(t))\tilde{g}_t(x_t). \tag{7}$$

Recall that, in addition to controlling the CCV, we also want to minimize the cumulative cost (which is equivalent to the regret minimization). Inspired by the stochastic *drift-plus-penalty* framework of Neely (2010), we combine these two objectives to a single objective of minimizing a sequence of surrogate cost functions $\{\hat{f}_t\}_{t=1}^T$ which is obtained by taking a positive linear combination of the drift upper bound (7) and the cost function. More precisely, we define

$$\hat{f}_t(x) := V\tilde{f}_t(x) + \Phi'(Q(t))\tilde{g}_t(x), \quad t \geq 1. \tag{8}$$

In the above, V is a suitably chosen non-negative parameter to be fixed later. Our proposed policy for COCO, described in Algorithm 1, simply runs the AdaGrad algorithm on the surrogate cost function sequence $\{\hat{f}_t\}_{t\geq 1}$, for a specific choice of $\Phi(\cdot)$ and V as dictated by the following analysis.

4.3.2. The Regret Decomposition Inequality

Let $x^\star \in \mathcal{X}^\star$ be any feasible action as defined in (1). Using the drift inequality from Eqn. (7), the definition of surrogate costs (8), and the fact that $g_\tau(x^\star) \leq 0, \forall \tau \geq 1$, we have

$$\begin{aligned} &\Phi(Q(\tau)) - \Phi(Q(\tau-1)) + V(\tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^\star)) \\ \leq\ & \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^\star),\ \forall \tau \geq 1. \end{aligned}$$

Summing up the above inequalities for $1 \leq \tau \leq t$, and using the fact that $\Phi(0) = 0$, we obtain

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \text{Regret}'_t(x^\star),\ \forall x^\star \in \Omega^\star, \tag{9}$$

where Regret_t on the LHS and Regret'_t on the RHS of (9) refers to the regret for learning the pre-processed cost functions $\{\tilde{f}_t\}_{t\geq 1}$ and the surrogate cost functions $\{\hat{f}_t\}_{t\geq 1}$ respectively. Since Algorithm 1 uses the AdaGrad algorithm for learning the surrogate cost functions, from (5), we need to upper bound the gradients of the surrogate functions to derive the regret bound. Towards this, the ℓ_2-norm of the gradients G_t of the surrogate cost function $\hat{f}_t$ can be bounded using the triangle inequality as follows:

$$\begin{aligned} G_t &\leq V||\nabla \tilde{f}_t(x_t)||_2 + \Phi'(Q(t))||\nabla \tilde{g}_t(x_t)||_2 \\ &\leq (2D)^{-1}\big(V + \Phi'(Q(t))\big). \end{aligned} \tag{10}$$

where in the last step, we have used the fact that the pre-processed functions are $(2D)^{-1}$-Lipschitz. Hence, plugging in the adaptive regret bound (5) on the RHS of (9), we arrive at the following generalized regret decomposition inequality valid for any $t \geq 1$:

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \sqrt{\sum_{\tau=1}^t \big(\Phi'(Q(\tau))\big)^2} + V\sqrt{t}. \tag{11}$$

where we have utilized simple algebraic inequalities $(x + y)^2 \leq 2(x^2 + y^2)$ and $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}, a, b \geq 0$.

Finally, recall that the sequence $\{Q(t)\}_{t\geq 1}$ is non-negative and non-decreasing as $\tilde{g}_t \geq 0$. Furthermore, the derivative $\Phi'(\cdot)$ is non-decreasing as the function Φ is assumed to be convex. Hence, upper-bounding all terms in the summation of the RHS of (11) by the last term, we arrive at the following simplified bound

$$\Phi(Q(t)) + V\text{Regret}_t(x^\star) \leq \Phi'\big(Q(t)\big)\sqrt{t} + V\sqrt{t}. \tag{12}$$

The regret decomposition inequality (12) constitutes the key step for the subsequent analysis.

4.3.3. Analysis

An Exponential Lyapunov function: We now derive the Regret and CCV bounds for the proposed policy with an exponential Lyapunov function $\Phi(x) \equiv \exp(\lambda x) - 1$, for

some parameter $\lambda \geq 0$ to be fixed later. An analysis with a power-law potential function is given in Appendix B, which also yields similar bounds. It is easy to verify that the function $\Phi(\cdot)$ satisfies required conditions- it is a non-decreasing and convex function with $\Phi(0) = 0$.

Bounding the Regret: From (12), for any feasible $x^\star \in \mathcal{X}^\star$ and for any $t \in [T]$, we have

$$\exp(\lambda Q(t)) + V\text{Regret}_t(x^*) \leq \lambda \exp(\lambda Q(t))\sqrt{t} + V\sqrt{t}.$$

Transposing, this yields the following bound on Regret:

$$\text{Regret}_t(x^\star) \leq \sqrt{t} + \frac{1}{V} + \frac{\exp(\lambda Q(t))}{V}(\lambda\sqrt{t} - 1). \quad (13)$$

Choosing any $\lambda \leq \frac{1}{\sqrt{T}}$, the last term in the above inequality becomes non-positive for any $t \in [T]$. Hence, for any $x^\star \in \mathcal{X}^\star$, we have

$$\text{Regret}_t(x^\star) \leq \sqrt{t} + \frac{1}{V}. \ \forall t \in [T]. \quad (14)$$

Bounding the CCV: Since all pre-processed cost functions are $(2D)^{-1}$-Lipschitz, we trivially have $\text{Regret}_t(x^\star) \geq -\frac{Dt}{2D} \geq -\frac{t}{2}$. Hence, from Eqn. (13), we have for any $\lambda < \frac{1}{\sqrt{T}}$ and any $t \in [T]$:

$$\begin{aligned} &\frac{\exp(\lambda Q(t))}{V}(1 - \lambda\sqrt{t}) \leq 2t + \frac{1}{V} \\ &\implies Q(t) \leq \frac{1}{\lambda} \ln \frac{1 + 2Vt}{1 - \lambda\sqrt{t}}. \end{aligned} \quad (15)$$

Finally, setting $\lambda = \frac{1}{2\sqrt{T}}$ and $V = 1$, and scaling the bounds back by $\alpha^{-1} \equiv 2GD$, we arrive at our main result.

Theorem 4.2. *For the COCO problem with adversarially chosen G-Lipschitz cost and constraint functions, Algorithm 1 yields the following Regret and CCV bounds for any horizon length $T \geq 1$:*

$$\begin{aligned} \textit{Regret}_t &\leq 2GD(\sqrt{t} + 1), \ \forall t \in [T] \\ \textit{CCV}_T &\leq 4GD \ln(2\big(1 + 2T\big))\sqrt{T}. \end{aligned}$$

In the above, D denotes the Euclidean diameter of the closed and convex admissible set $\mathcal{X}$.

Remarks: Although, in the above, we assume that the horizon length T is known, we can use the standard doubling trick for unknown T.

In the final section, we show that in addition to a small regret and CCV bounds, the proposed Algorithm 1 also has a small movement cost measured in the (squared) Euclidean metric. These features make it attractive for practical applications.

4.4. Bounding the movement cost

The movement cost of the AdaGrad policy can be bounded as

$$\begin{aligned} ||x_{t+1} - x_t||_2^2 &\overset{(a)}{\leq} ||\eta_t \nabla_t||_2^2 = \frac{D^2}{2} \frac{G_t^2}{\sum_{\tau=1}^t G_\tau^2} \\ &\leq \frac{D^2}{2} \int_{\sum_{\tau=1}^{t-1} G_\tau^2}^{\sum_{\tau=1}^t G_\tau^2} \frac{dx}{x}, \end{aligned}$$

where in step (a), we have used the Pythagorean theorem for Euclidean projection (Hazan, 2022, Theorem 4). Summing up, the cumulative squared-Euclidean movement cost of the AdaGrad policy for T rounds can be bounded as

$$\sum_{t=1}^T ||x_{t+1} - x_t||_2^2 \leq \frac{D^2}{2}\big(1 + \ln(\sum_{\tau=1}^T G_\tau^2) - \ln G_1^2\big). \quad (16)$$

Equation (16) gives an upper bound to the movement cost for the AdaGrad policy for any sequence of cost functions.

4.4.1. Movement cost of Algorithm 1

Since Algorithm 1 uses the AdaGrad policy on the sequence of surrogate cost functions, we can use Eqn. (16) for bounding its total movement cost. Recall that, Eqn. (10) gives an upper bound to the gradient of the surrogate cost functions fed to the AdaGrad policy. Hence, we have

$$\begin{aligned} \sum_{\tau=1}^T G_\tau^2 &\leq G^2 \sum_{\tau=1}^T (V + \Phi'(Q(\tau))^2 \\ &\overset{(a)}{\leq} 2G^2T(1 + \lambda^2 \exp(2\lambda Q(T))) \\ &\overset{(b)}{\leq} 4G^2T(3 + 2T), \end{aligned} \quad (17)$$

where in (a), we have used the inequality $(x + y)^2 \leq 2(x^2 + y^2)$, substituted the value of the parameter $V = 1$ and used the fact that $\{Q(t\}_{t\geq 1}$ is a non-decreasing sequence. In (b), we have used the bound from (15) and substituted for the parameter $\lambda = 1/(2\sqrt{T})$. Substituting the bound (17) into (16), the squared-Euclidean movement cost for Algorithm 1 can be bounded as $\sum_{t=1}^T ||x_{t+1} - x_t||_2^2 = O((\log T))$. Using the Cauchy-Schwarz inequality, the Euclidean movement cost can also be bounded as $\sum_{t=1}^T ||x_{t+1} - x_t||_2 \leq \sqrt{T}\sqrt{\sum_{t=1}^T ||x_{t+1} - x_t||_2^2} = \tilde{O}(\sqrt{T})$.

5. Conclusion

In this paper, we proposed an efficient online policy that achieves the optimal regret and CCV bounds for the COCO problem without any assumptions. We have also shown that the squared Euclidean movement cost of the proposed policy is small and is bounded by $O((\log T)^2)$. These results are established by using a new proof technique involving an exponential Lyapunov function.

6. Acknowledgement

The work was supported in part by a Google India faculty Research award. The first author was additionally supported by an NSF-DST collaborative grant coordinated by IDEAS-Technology Innovation Hub (TIH) at the Indian Statistical Institute, Kolkata.

References

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Guo, H., Liu, X., Wei, H., and Ying, L. Online convex optimization with hard constraints: Towards the best of two worlds and beyond. *Advances in Neural Information Processing Systems*, 35:36426–36439, 2022.

Hazan, E. *Introduction to online convex optimization*. MIT Press, 2022.

Jenatton, R., Huang, J., and Archambeau, C. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pp. 402–411. PMLR, 2016.

Mahdavi, M., Jin, R., and Yang, T. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1):2503–2528, 2012.

Neely, M. J. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

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

Orabona, F. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Sinha, A. BanditQ: Fair bandits with guaranteed rewards. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

Sinha, A. and Vaze, R. Playing in the dark: No-regret learning with adversarial constraints, 2023.

Yi, X., Li, X., Yang, T., Xie, L., Chai, T., and Johansson, K. Regret and cumulative constraint violation analysis for online convex optimization with long term constraints. In *International Conference on Machine Learning*, pp. 11998–12008. PMLR, 2021.

Yi, X., Li, X., Yang, T., Xie, L., Hong, Y., Chai, T., and Johansson, K. H. Distributed online convex optimization with adversarial constraints: Reduced cumulative constraint violation bounds under slater's condition. *arXiv preprint arXiv:2306.00149*, 2023.

Yu, H. and Neely, M. J. A low complexity algorithm with $\mathcal{O}(\sqrt{T})$ regret and $\mathcal{O}(1)$ constraint violations for online convex optimization with long term constraints. *Journal of Machine Learning Research*, 21(1):1–24, 2020.

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

Yuan, J. and Lamperski, A. Online convex optimization for cumulative constraints. *Advances in Neural Information Processing Systems*, 31, 2018.

A. Appendix

Table 1. Summary of the results for the COCO problem with convex cost and convex constraint functions. Unless specified otherwise in the Assumptions, both the cost and constraint functions are assumed to be chosen by an adaptive adversary. Details of the abbreviations appearing on the table can be found in the trailing note.

REFERENCE	REGRET	CCV	COMPLEXITY PER ROUND	ASSUMPTIONS
MAHDAVI ET AL. (2012)	$O(\sqrt{T})$	$O(T^{3/4})$	PROJ.	FIXED CONSTR.
JENATTON ET AL. (2016)	$O(T^{\max(\beta,1-\beta)})$	$O(T^{1-\beta/2})$	PROJ	FIXED CONSTR.
YU ET AL. (2017)	$O(\sqrt{T})$	$O(\sqrt{T})$	PROJ	SLATER & STOCH.
NEELY & YU (2017)	$O(\sqrt{T})$	$O(\sqrt{T})$	PROJ.	SLATER
YU & NEELY (2020)	$O(\sqrt{T})$	$O(1)$	CONV-OPT	SLATER & FIXED CONSTR.
YI ET AL. (2021)	$O(\sqrt{T})$	$O(T^{1/4})$	CONV-OPT	FIXED CONSTR.
GUO ET AL. (2022)	$O(\sqrt{T})$	$O(T^{3/4})$	CONV-OPT	—
YI ET AL. (2023)	$O(T^{\max(\beta,1-\beta)})$	$O(T^{1-\beta})$	CONV-OPT	SLATER
SINHA & VAZE (2023)	$O(\sqrt{T})$	$O(T^{3/4})$	PROJ	—
SINHA & VAZE (2023)	$O(\sqrt{T})$	$O(\sqrt{T})$	PROJ	$\text{REGRET}_T \geq 0$
THIS PAPER	$O(\sqrt{T})$	$\tilde{O}(\sqrt{T})$	PROJ	—

Notes:

1. CONV-OPT: Solution of a convex optimization problem over the convex constraint set $\mathcal{X}$
2. PROJ: Euclidean projection onto the set $\mathcal{X}$
3. PROJ operation is computationally much cheaper than CONV-OPT for typical $\mathcal{X}$.
4. SLATER: Slater condition
5. STOCH: Stochastic constraints, *i.e.,* the constraint functions are generated i.i.d. per slot.
6. FIXED CONSTR.: Time-invariant fixed constraint functions known to the algorithm. Here the goal is to avoid the complex projection step onto the constraint set.
7. β is any fixed parameter in the interval $[0, 1]$

A.1. Details on the assumptions made in the previous papers

Slater's condition: This condition assumes that there exists a positive $\epsilon > 0$ and a feasible action $x \in \mathcal{X}$ such that $g_t(x) \leq -\epsilon, \forall t \in [T]$. Trivially, Slater's condition does not hold for clipped constraints, *i.e.*, constraints of the form $g_t(x) \equiv \max(0, \tilde{g}_t(x))$, for some convex function $\tilde{g}_t$. Worse yet, upon assuming Slater's condition, the CCV bounds derived in Neely & Yu (2017, Theorem 4 (b)), Yu et al. (2017, Theorem 1), Yu & Neely (2020), and Yi et al. (2023, Theorem 2) diverge to infinity as the slack ϵ approaches zero. This should be compared against our main result in Theorem 4.2, where the bounds remain finite for any finite G, D, and T while making no assumption on Slater's condition.

Non-negativity of Regret: Since the benchmark action in the regret definition (2) belongs to the feasible set $\mathcal{X}^\star$, it is computationally non-trivial to determine whether the regret is non-negative or not (Sinha & Vaze, 2023). While it is shown in Sinha & Vaze (2023, Theorem 9), that the improved $O(\sqrt{T})$ CCV bound holds for convex adversaries and fixed constraints, improving the CCV bound for the general case with adversarially chosen constraints and arbitrary adversaries was left as an open problem.

B. Analysis with A Power-law Lyapunov Function

We now specialize inequality (12) by considering the power-law Lyapunov function $\Phi(x) \equiv x^n$ for some integer $n \geq 2$, to be fixed later. The pseudocode for the policy remains the same as in Algorithm 1 with the new set of parameters $\alpha = \frac{1}{2GD}, n = \max(2, \lceil \ln T \rceil), V = (n-1)^{n-1} T^{\frac{n-1}{2}}, \Phi(x) = x^n$.

From (12), we obtain

$$Q^n(t) + V\text{Regret}_t(x^\star) \leq nQ^{n-1}(t)\sqrt{t} + V\sqrt{t}. \tag{18}$$

Bounding the Regret: From Eqn. (18), we have the following regret bound for any feasible $x^\star \in \mathcal{X}^\star$:

$$\begin{aligned}\text{Regret}_t(x^\star) &\leq \sqrt{t} + \frac{Q^{n-1}(t)}{V}(n\sqrt{t} - Q(t)) \\ &\leq \sqrt{t} + \frac{(n-1)^{n-1}t^{n/2}}{V},\end{aligned}$$

where the last step follows from the AM-GM inequality. Finally, upon taking $V = (n-1)^{n-1}T^{\frac{n-1}{2}}$ as in Algorithm 1, we obtain $\text{Regret}_t(x^\star) \leq 2\sqrt{t},\ \forall 1 \leq t \leq T$.

Bounding the CCV: Since all pre-processed cost functions are $(2D)^{-1}$-Lipschitz, we trivially have $\text{Regret}_t(x^\star) \geq -{}^{Dt}/_{2D} \geq -\frac{t}{2}$. Hence, from Eqn. (18), we obtain

$$Q^n(t) \leq 2Vt + nQ^{n-1}(t)\sqrt{t}. \tag{19}$$

Finally, to bound $Q(t)$ from the above, consider the case when $Q(t) \geq 2n\sqrt{t}$. In this case, from (19), we have

$$Q^n(t) \leq 2Vt + \frac{1}{2}Q^n(t) \implies Q(t) \leq (4Vt)^{\frac{1}{n}}.$$

Thus, in general, the following bound holds

$$Q(t) \leq \max(2n\sqrt{t}, (4Vt)^{\frac{1}{n}}).$$

Substituting the chosen value for the parameter V and using the fact that $n \geq 2$, from the above, we have

$$Q(t) \leq \max(2n\sqrt{T}, 2nT^{\frac{1}{2}+\frac{1}{2n}}) = 2nT^{\frac{1}{2}+\frac{1}{2n}},\ \forall t \in [T]. \tag{20}$$

Finally, setting $n = \max(2, \lceil \ln T \rceil)$ as in Algorithm 1, we obtain the following CCV bound:

$$\text{CCV}_T = Q(T) \leq 3.3(2 + \ln T)\sqrt{T},\ \forall T \geq 1.$$

Recall that the above results hold for the pre-processed functions. Hence, scaling the bounds back by $\alpha^{-1} \equiv 2GD$, we conclude that the proposed policy yields the following Regret and CCV bounds

$$\boxed{\begin{aligned}\text{Regret}_t &\leq 4GD\sqrt{T},\ \ \forall t \in [T] \\ \text{CCV}_T &\leq 6.6GD(2 + \ln T))\sqrt{T}.\end{aligned}}$$

Bounding the Movement cost As before, we only need to bound the quantity $\sum_{\tau=1}^{T} G_\tau^2$. We have

$$\begin{aligned}\sum_{\tau=1}^{T} G_\tau^2 &\leq G^2 \sum_{\tau=1}^{T} (V + \Phi'(Q(\tau))^2 \\ &\leq 2G^2(V^2T + n^2 \sum_{\tau=1}^{T} Q^{2n-2}(\tau)) \\ &\overset{(a)}{\leq} 2G^2(n^{2n-2}T^n + (2n)^{2n-2}T^n) \\ &\leq G^2(2n)^{2n}T^n.\end{aligned}$$

where in (a), we have substituted the value of the parameter V and used the bound from Eqn. (20). Finally, substituting the above in (16), we conclude that the squared-Euclidean movement cost for Algorithm 1 can be bounded as

$$\sum_{t=1}^{T} ||x_{t+1} - x_t||_2^2 = O((\log T)^2),$$

which is slightly larger than the logarithmic movement cost with the exponential Lyapunov function. Using the Cauchy-Schwarz inequality, its Euclidean movement cost can also be easily bounded as

$$\sum_{t=1}^{T} ||x_{t+1} - x_t||_2 \leq \sqrt{T}\sqrt{\sum_{t=1}^{T} ||x_{t+1} - x_t||_2^2} = \tilde{O}(\sqrt{T}).$$