Optimal Anytime Algorithms for Online Convex Optimization with Adversarial Constraints

Dhruv Sarkar[1], Abhishek Sinha[2]

[1] Indian Institute of Technology Kharagpur
[2] Tata Institute of Fundamental Research Mumbai
dhruv.sarkar@kgpian.iitkgp.ac.in, abhishek.sinha.tifr@gmail.com

Abstract

We propose an anytime online algorithm for the problem of learning a sequence of adversarial convex cost functions while approximately satisfying another sequence of adversarial online convex constraints. A sequential algorithm is called *anytime* if it provides a non-trivial performance guarantee for any intermediate timestep t without requiring prior knowledge of the length of the entire time horizon T. Our proposed algorithm achieves optimal performance bounds without resorting to the standard doubling trick, which has poor practical performance due to multiple restarts. Our core technical contribution is the use of time-varying Lyapunov functions to keep track of constraint violations. This must be contrasted with prior works that used a fixed Lyapunov function tuned to the known horizon length T. The use of time-varying Lyapunov function poses unique analytical challenges as properties, such as *monotonicity*, on which the prior proofs rest, no longer hold. By introducing a new analytical technique, we show that our algorithm achieves $O(\sqrt{t})$ regret and $\tilde{O}(\sqrt{t})$ cumulative constraint violation bounds for any $t \geq 1$.

We extend our results to the dynamic regret setting, achieving bounds that adapt to the path length of the comparator sequence without prior knowledge of its total length. We also present an adaptive algorithm in the optimistic setting, whose performance gracefully scales with the cumulative prediction error. We demonstrate the practical utility of our algorithm through numerical experiments involving the online shortest path problem.

1 Introduction

We consider a natural generalization of the standard Online Convex Optimization (OCO) problem, called Constrained Online Convex Optimization (COCO), where in addition to a convex cost function, a convex constraint function is revealed to the learner after it chooses its action at each round for a horizon of T rounds. The goal is to design an online algorithm which achieves a small regret (w.r.t. the cost functions) while ensuring a small cumulative constraint violation or CCV (w.r.t. the constraint functions) for all rounds $t \geq 1$. The COCO problem arises in many practical settings, including safety-aware contextual bandits (Sun, Dey, and Kapoor 2017), autonomous driving (Gao et al. 2024), budget-

constrained bandits (Immorlica et al. 2022), and learning safety constraints in LLMs (Chen, As, and Krause 2025).

The COCO problem has been extensively studied in the literature over the past decade (Guo et al. 2022; Yi et al. 2023; Neely and Yu 2017). Recent work by Sinha and Vaze (2024) introduced a simple algorithm which yields $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV guarantees, which are also shown to be tight when the horizon length T is known. These results are established by leveraging a Lyapunov function-based technique that constructs a surrogate cost function by linearly combining the cost and constraint functions at each round, and then passes it on to a standard OCO algorithm. The analysis is also refreshingly elegant in stark contrast to prior works that rely on primal-dual-based approaches and achieve only suboptimal guarantees (Guo et al. 2022; Yi et al. 2021, 2023; Sun, Dey, and Kapoor 2017).

However, a major drawback of their result is that their proposed algorithm is *not* anytime as it needs to know the length of the horizon T *a priori*. In particular, although their algorithm achieves $O(\sqrt{t})$ regret for any $t \geq 1$, the CCV guaranteed by their algorithm is not time-uniform as it only achieves a CCV of $\tilde{O}(\sqrt{T})$ for any time $t \geq 1$, however small (Sinha and Vaze 2024, Theorem 1). This raises a natural and important question - is it possible to design a COCO algorithm which simultaneously achieves $O(\sqrt{t})$ regret and $\tilde{O}(\sqrt{t})$ CCV for any $t \geq 1$?

In this paper, we affirmatively answer this question by introducing a sequence of time-varying Lyapunov functions $\{\Phi_t(\cdot)\}_{t \geq 1}$, which replace the fixed, horizon-dependent Lyapunov function $\Phi(\cdot)$ employed by Sinha and Vaze (2024). Their analysis crucially utilized the monotone *non-decreasing* property of the $\{\Phi(Q(t))\}_{t \geq 1}$ sequence, where $Q(t)$ is the CCV up to round t. Unfortunately, it turns out that with the new Lyapunov functions, $\Phi_t(x)$ is actually *monotonically decreasing* pointwise in t, and hence, the previous arguments do not apply. The key technical novelty in our analysis is in ensuring that the sequence $\{\Phi_t(Q(t))\}_{t \geq 1}$ remains monotonically non-decreasing even though $\Phi_t(x) < \Phi_{t-1}(x), \forall x$. This is achieved by defining $Q(t)$ to be a suitable upper bound to the current CCV while adequately exploiting the exponential form of the Lyapunov functions (see Eqn. (6)).

In summary, we make the following contributions in the paper:

- We propose a simple anytime algorithm for COCO that achieves $O(\sqrt{t})$ regret and $\tilde{O}(\sqrt{t})$ CCV for any $t \geq 1$. Our algorithm does not need multiple restarts and completely avoids the impractical doubling trick (see section A for more details).

- The above results are obtained by introducing a sequence of time-varying Lyapunov functions in both our algorithms and their analysis. Compared to the prior works, which use time-invariant Lyapunov functions, time-varying Lyapunov functions pose non-trivial technical challenges, as the key monotonicity property is lost. We address this challenge by working with a suitable upper bound to CCV and appropriately tuning the parameters of the Lyapunov functions.

- We show that our techniques extend naturally to the dynamic regret and optimistic settings. In particular, we design algorithms with performance bounds that adapt to unknown parameters, such as the total path length $\mathcal{P}_T$ or the total prediction error $\mathcal{E}_T$, without resorting to the practically unwieldy doubling trick.

- Finally, we demonstrate the superiority of our algorithm through a number of numerical experiments.

2 Related Works

Online learning with long-term constraints was first studied by Mannor, Tsitsiklis, and Yu (2009). In the context of two-player infinite-horizon stochastic games, they established a fundamental impossibility result: it is not possible to simultaneously achieve sublinear bounds for both regret as well as cumulative constraint violation (CCV) against the best fixed offline action that satisfies the long-term constraint over the entire horizon. This negative result motivated subsequent works to consider a weaker benchmark, in which the benchmark satisfies the constraints at *every* round (Mahdavi, Jin, and Yang 2012; Neely and Yu 2017; Guo et al. 2022). The goal in this line of work is to obtain the tightest regret and CCV bounds under this assumption. For time-invariant constraints, Mahdavi, Jin, and Yang (2012) were the first to use Online Gradient Descent (OGD) and mirror prox-based online policies to obtain sublinear regret and sublinear CCV guarantees. Later, Castiglioni et al. (2022) proposed a unified meta-algorithm that achieves $\mathcal{O}(T^{3/4})$ bounds for both approximate regret and CCV in the non-convex setting with long-term constraints. However, they made an additional *Slater's condition* (strict feasibility) assumption and the assumption that the constraint functions change no faster than $O(T^{-1/4})$. Guo et al. (2022) studied the problem with adversarial constraints, and achieved $O(\sqrt{T})$ regret and $O(T^{3/4})$ CCV in the general setting without assuming Slater's condition. The closest to our work is the paper by Sinha and Vaze (2024), which achieves optimal $O(\sqrt{T})$ regret and $\tilde{O}(\sqrt{T})$ CCV bounds by reducing the problem to the standard online convex optimization on a sequence of surrogate cost functions. However, they assumed apriori knowledge of the horizon T and also their violation bound was suboptimal in the sense that even for an intermediate timestep t, they could only establish $\tilde{O}(\sqrt{T})$ bound as opposed to the $\tilde{O}(\sqrt{t})$

that we present. This was primarily because the definition of the surrogate costs required the knowledge of the horizon length T. Follow-up works, such as Lekeufack and Jordan (2025), which generalized Sinha and Vaze (2024)'s algorithm to the optimistic and dynamic regret setting, required the knowledge of other problem-dependent parameters, such as path length or error bounds, which prevented their algorithms from being truly adaptive. All of the above papers relied on the doubling trick to estimate these problem dependent parameters. However, as noted in a number of prior works, the doubling trick is impractical and wasteful as it relies on restarting the algorithm from scratch every time the problem dependent parameter exceeds the estimate (Zhang et al. 2024; Kwon and Mertikopoulos 2014; Luo and Schapire 2014). Consequently, the data from the previous learning phases are not utilized for prediction in the subsequent phase, which leads to unsatisfactory practical performance. We have discussed the doubling trick in more detail in Section A in the Appendix.

3 Problem formulation

We consider a repeated game between an online player and an adaptive adversary. In this game, on each round $t \geq 1$, the player chooses a feasible action x_t from a decision set $\mathcal{X}$. The set $\mathcal{X}$ is assumed to be non-empty, closed, and convex with a finite Euclidean diameter of D. Upon observing the current action x_t, the adversary chooses two convex and G-Lipschitz 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 corresponds 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 player every round. Consequently, the player incurs a cost of $f_t(x_t)$ and an instantaneous constraint violation of $\max(0, g_t(x_t))$. Our objective is to design an online policy that achieves a small cumulative cost while incurring a small CCV for each $t \geq 1$. The (static) 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 round. Specifically, let $\mathcal{X}^\star \subseteq \mathcal{X}$ be the set consisting of all actions satisfying all constraints:

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

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

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

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

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, $\mathcal{X}^\star = \mathcal{X}$.

4 Anytime bounds on Regret and Cumulative Constraint Violation

4.1 Technical Overview

Our approach builds upon the Lyapunov-based framework recently proposed by Sinha and Vaze (2024). They reduce the COCO problem to an instance of the standard OCO problem on a sequence of surrogate cost functions that linearly combine the cost and constraint functions at each round. The coefficients of the linear combinations are chosen according to a fixed Lyapunov function $\Phi(x) := e^{\lambda x} - 1$, where the parameter λ depends on the length of the time horizon T. Hence, their algorithm is not anytime and the resulting bounds are not time-uniform. To overcome this, we use time-varying Lyapunov functions, $\Phi_t(x) = e^{\lambda_t x} - 1$, where the parameter λ_t is a decreasing function of the current time step t.

Although the time-varying Lyapunov functions obviates the need to know the horizon length T, their analysis presents a significant challenge: the sequence $\Phi_t(Q(t))$ is no longer guaranteed to be monotonically non-decreasing, a property that was critical to the analysis in Sinha and Vaze (2024). Our key technical novelty is to restore this monotonicity by redefining the variables $Q(t)$. Instead of the standard additive update, which leads to an exact expression for CCV, we introduce a multiplicative factor, defining $Q(t) = \frac{\lambda_{t-1}}{\lambda_t}Q(t-1) + \tilde{g}_t(x_t)$. Since λ_t is decreasing, the factor $\frac{\lambda_{t-1}}{\lambda_t} \geq 1$ precisely compensates for the shrinking Lyapunov function, ensuring that $\Phi_t(Q(t)) \geq \Phi_{t-1}(Q(t-1))$. Furthermore, the new definition also ensures that $Q(t)$ remains an upper bound to the CCV for any $t \geq 1$.

Finally, we construct a surrogate cost function $\hat{f}_t$ based on the time-varying Lyapunov functions, which are then passed to a standard adaptive OCO algorithm like AdaGrad. This procedure allows us to derive the desired anytime bound of $O(\sqrt{t})$ for regret and $\tilde{O}(\sqrt{t})$ for cumulative constraint violation.

4.2 Preliminaries: Lipschitz-Adaptive OCO Algorithms

Algorithm 1: AdaGrad: Online Gradient Descent with Adaptive step sizes

---

1: **Input** : Convex decision set $\mathcal{X}$, sequence of convex cost functions $\{\hat{f}_t\}_{t=1}^T$, sequence of learning rates $\{\eta_t\}_{t=1}^T$, diam$(\mathcal{X}) = D$, $\mathcal{P}_{\mathcal{X}}(\cdot)$ = Euclidean projection operator on the convex set $\mathcal{X}$.
2: **Initialize** : $x_1 \in \mathcal{X}$ arbitrarily.
3: **for** $t = 1 : T$ **do**
4:     Play x_t and compute $\nabla_t = \nabla \hat{f}_t(x_t)$
5:     Set $x_{t+1} = \mathcal{P}_{\mathcal{X}}(x_t - \eta_t \nabla_t)$
6: **end for**

In this Section, we briefly recall the family of first-order methods (*a.k.a.* Projected Online Gradient Descent (OGD)) for the standard OCO problem, which will be used as a subroutine in our proposed COCO algorithm. These methods differ from 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 an adaptive step size sequence. The (diagonal version of the) AdaGrad policy adaptively chooses the step size sequence as a function of the norm 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, Hazan, and Singer 2011). [1] Importantly, the AdaGrad policy does not need to know the horizon length T or a uniform upper bound to the Lipschitz constants of the cost functions. This policy enjoys the following adaptive regret bound.

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

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

4.3 Design and Analysis of the COCO 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 functions are obtained upon multiplying the bounds by α^{-1}.

Surrogate cost functions: Let $\{\lambda_t\}_{t \geq 1}$ be a monotonically decreasing sequence which will be specified later. We define $Q(t)$ using the following recursion:

$$Q(t) = \frac{\lambda_{t-1}}{\lambda_t}Q(t-1) + \tilde{g}_t(x_t), \ t \geq 1, \tag{6}$$

with $Q(0) = 0$. Since $\frac{\lambda_{t-1}}{\lambda_t} \geq 1$, $Q(t)$ is an upper bound to the CCV up to round t. Thus, an upper bound to $Q(t)$ also implies an upper bound to CCV. Consequently, we proceed to control $Q(t)$.

Towards this, let $\Phi_t : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be a sequence of non-decreasing convex Lyapunov functions such that $\Phi_0(0) = 0$. Using the convexity of the function $\Phi_t(\cdot)$, we have

$$\Phi_t(Q(t)) \leq \Phi_t(\frac{\lambda_{t-1}}{\lambda_t}Q(t-1)) + \Phi_t'(Q(t))\tilde{g}_t(x_t). \tag{7}$$

---

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

Algorithm 2: Anytime Online Policy for COCO

1: **Input:** 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 $\mathcal{X}$

2: **Parameters:** $\lambda_t = \frac{1}{4\sqrt{t}\sqrt{\log t + 1}(\log(\log t + 1) + 1)}, \Phi_t(x) = \exp(\lambda_t x) - 1, \alpha = (2GD)^{-1}$.

3: **Initialization:** Set $x_1 = 0, Q(0) = 0$.

4: **for** $t = 1 : T$ **do**

5:    Choose x_t, observe f_t, g_t, incur a cost of $f_t(x_t)$ and constraint violation of $(g_t(x_t))^+$

6:    $\tilde{f}_t \leftarrow \alpha f_t, \tilde{g}_t \leftarrow \alpha \max(0, g_t)$.

7:    $Q(t) = \frac{\lambda_{t-1}}{\lambda_t} Q(t-1) + \tilde{g}_t(x_t)$.

8:    Compute $\hat{f}_t$ as per (9)

9:    Pass $\hat{f}_t$ and $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{\tau=1}^t \|\nabla_\tau\|_2^2}}$ to Algorithm 1.

10: **end for**

We now choose the Lyapunov function to be $\Phi_t(x) \overset{\text{(def)}}{=} e^{\lambda_t x} - 1$. Hence, we have $\Phi_t(\frac{\lambda_{t-1}}{\lambda_t} x) = \Phi_{t-1}(x), \forall x$. Substituting this in the bound (7), the one-step change (*drift*) of the potential function $\Phi_t(Q(t))$ can be upper bounded as

$$\Phi_t(Q(t)) - \Phi_{t-1}(Q(t-1)) \le \Phi_t'(Q(t))\tilde{g}_t(x_t). \quad (8)$$

Recall that, in addition to controlling the CCV, we also want to minimize the cumulative cost (which is equivalent to minimizing the regret for the cost functions). We combine these two objectives into a single objective of minimizing a sequence of surrogate cost functions $\{\hat{f}_t\}_{t=1}^T$ obtained by taking a positive linear combination of the drift upper bound (8) and the cost function f_t. More precisely, we define the surrogate cost function at round t as

$$\hat{f}_t(x) := \tilde{f}_t(x) + \Phi_t'(Q(t))\tilde{g}_t(x), \quad t \ge 1. \quad (9)$$

Our proposed policy for COCO, described in Algorithm 2, simply runs the AdaGrad algorithm on the surrogate cost function sequence $\{\hat{f}_t\}_{t\ge 1}$, for a specific choice of the parameter sequence $\{\lambda_t\}_{t\ge 1}$ as dictated by the following analysis.

The Regret Decomposition Inequality   Let $x^\star \in \mathcal{X}^\star$ be any feasible benchmark action as defined by Eqn. (1). Using the drift inequality from Eqn. (8), the definition of surrogate costs (9), and the fact that $g_\tau(x^\star) \le 0, \forall \tau \ge 1$, we have

$$\Phi_\tau(Q(\tau)) - \Phi_{\tau-1}(Q(\tau-1)) + (\tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^\star))$$
$$\le \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^\star), \forall \tau \ge 1.$$

Summing up the above inequalities for $1 \le \tau \le t$, and using the fact that $\Phi_0(0) = 0$, we obtain

$$\Phi_t(Q(t)) + \text{Regret}_t(x^\star) \le \text{Regret}_t'(x^\star), \forall x^\star \in \Omega^\star, \quad (10)$$

where Regret_t on the LHS and Regret_t' on the RHS of (10) refers to the regret for learning the pre-processed cost functions $\{\tilde{f}_t\}_{t\ge 1}$ and the surrogate cost functions $\{\hat{f}_t\}_{t\ge 1}$ respectively (see Eqn. (2)). From Eqn. (5), the regret bound

for the AdaGrad algorithm depends on the ℓ_2 norms of the gradients of the input cost functions. Since we use AdaGrad for learning the surrogate cost functions $\{\hat{f}_t\}_{t\ge 1}$, we need to upper bound the gradients of the surrogate functions to derive the regret expression. 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:

$$
\begin{aligned}
G_t &\le \|\nabla \tilde{f}_t(x_t)\|_2 + \Phi_t'(Q(t))\|\nabla \tilde{g}_t(x_t)\|_2 \\
&\le (2D)^{-1}\big(1 + \Phi_t'(Q(t)\big). \quad (11)
\end{aligned}
$$

where in the last step, we have used the fact that the preprocessed functions are $(2D)^{-1}$-Lipschitz. Hence, plugging in the adaptive regret bound (5) on the RHS of (10), we arrive at the following regret decomposition inequality valid for any $t \ge 1$:

$$\Phi_t(Q(t)) + \text{Regret}_t(x^\star) \le \sqrt{t} + \sqrt{\sum_{\tau=1}^t \big(\Phi_\tau'(Q(\tau))\big)^2}. \quad (12)$$

In the above, we have utilized simple algebraic inequalities $(x + y)^2 \le 2(x^2 + y^2)$ and $\sqrt{a + b} \le \sqrt{a} + \sqrt{b}, a, b \ge 0$. Also note that, for our choice of Lyapunov function, $\Phi_\tau'(x) = \lambda_\tau e^{\lambda_\tau x} = \lambda_\tau(\Phi_\tau(x) + 1)$. Hence,

$$\sqrt{\sum_{\tau=1}^t \big(\Phi_\tau'(Q(\tau))\big)^2} \le \sqrt{\sum_{\tau=1}^t \lambda_\tau^2 \big(\Phi_\tau(Q(\tau)) + 1\big)^2}. \quad (13)$$

Further note that the recursion $Q(\tau) = \frac{\lambda_{\tau-1}}{\lambda_\tau} Q(\tau-1) + \tilde{g}_\tau(x_\tau)$ implies $\lambda_\tau Q(\tau) \ge \lambda_{\tau-1} Q(\tau - 1)$ as $\tilde{g}_\tau \ge 0$. Therefore, $\Phi_\tau(Q(\tau)) \ge \Phi_{\tau-1}(Q(\tau-1))$, *i.e.*, the sequence $\{\Phi_\tau(Q(\tau))\}_{\tau \ge 1}$ is non-negative and non-decreasing. Hence, upper-bounding all terms within the parentheses in the summation of the RHS of (13) by the last term, we arrive at the following simplified bound

$$\sqrt{\sum_{\tau=1}^t \lambda_\tau^2 \big(\Phi_\tau(Q(\tau)) + 1\big)^2} \le \sqrt{\sum_{\tau=1}^t \lambda_\tau^2(\Phi_t(Q(t)) + 1)}$$

Thus Eqn. (12) yields for any $t \ge 1$:

$$\Phi_t(Q(t)) + \text{Regret}_t(x^\star)$$
$$\le \sqrt{\sum_{\tau=1}^t \lambda_\tau^2 \Phi_t(Q(t))} + \sqrt{\sum_{\tau=1}^t \lambda_\tau^2} + \sqrt{t}. \quad (14)$$

The regret decomposition inequality (14) constitutes the key step for the subsequent analysis. We finally choose the parameters

$$\lambda_\tau = \frac{1}{4\sqrt{\tau}\sqrt{\log(\tau) + 1}(\log(\log(\tau) + 1) + 1)}, \tau \ge 1.$$

Note that $\sum_{\tau=1}^t \lambda_\tau^2 < \frac{1}{4}$ using simple calculus as given in Lemma B.1. This lets us conclude from (14) that,

$$\text{Regret}_t(x^\star) + \frac{1}{2}\Phi_t(Q(t)) \le \sqrt{t} + \frac{1}{2} \quad (15)$$

Noting that, $\Phi_t(Q(t)) \geq 0$, we get,

$$\text{Regret}_t(x^\star) \leq \sqrt{t} + 1$$

we trivially have $\text{Regret}_t(x^\star) \geq -\frac{Dt}{2D} \geq -\frac{t}{2}$. Hence, from Eqn. (15), we have,

$$\exp(\lambda_t Q(t)) \leq t + 2\sqrt{t} + 1 \leq 4t$$
$$\implies Q(t) \leq \frac{1}{\lambda_t} \log 4t$$
$$\implies Q(t) \leq 4\sqrt{t}\sqrt{\log t + 1}\left(\log(\log t + 1) + 1\right)\log 4t.$$

Further, noted above, since $\{\lambda_t\}_{t \in [T]}$ is a decreasing sequence, $\text{CCV}_t \leq Q(t)$ based on the definition of the $\{Q(t)\}$ sequence. The following theorem summarizes the above results

Theorem 4.2. *For the COCO problem with adversarially chosen G-Lipschitz cost and constraint functions, Algorithm 2 yields the following Regret and CCV bounds for any $t \geq 1$:*

$$\text{Regret}_t \leq 2GD(\sqrt{t} + 1),$$
$$\text{CCV}_t \leq 8GD\sqrt{t}\sqrt{\log t + 1}\left(\log(\log t + 1) + 1\right)\log 4t.$$

In the above, D denotes the diameter of the decision set $\mathcal{X}$.

Remark 4.3. Theorem 4.2 ensures a regret bound of $O(\sqrt{t})$ and a violation bound of $O(\sqrt{t}(\log t)^{3/2}\log\log t)$. With a slightly tighter analysis, one could improve the violation bound to $O(\sqrt{t}\log t \, (\prod_{k=1}^{\log^\star t} \log^{(k)} t)^{1/2})$ by choosing λ_τ appropriately. Here $\log^{(k)}$ represents the logarithm repeated k times while $\log^\star t$ represents the iterated logarithm.

Remark 4.4. We can derive an adaptive CCV bound of the form $\tilde{O}\big(\sqrt{\sum_{\tau=1}^{t}||\nabla g_t^+(x_t)||_2^2}\big)$ by choosing the value of λ_τ appropriately. Concretely, denoting $\gamma_t = \sum_{\tau=1}^{t}||\nabla g_t^+(x_t)||_2^2$ and taking $\lambda_\tau = \frac{1}{4\sqrt{\gamma_\tau}\sqrt{\log(\gamma_\tau)+1}\,(\log(\log(\gamma_\tau)+1)+1)}$ would suffice. This bound is tighter because it can adjust itself based on the problem instance, whereas the $\tilde{O}(\sqrt{t})$ bound was a worst-case bound which remained the same for all problem instances.

5 Extension to Dynamic Regret Bounds

While in the previous Section, our algorithm ensured a static regret guarantee against a fixed benchmark, in this Section, we show that our algorithmic technique also yields a dynamic regret guarantee against any sequence of time-varying benchmarks. More importantly, our algorithm does not need to know any upper bound on the worst-case path length corresponding to the benchmark sequence.

5.1 Adaptive OCO meta-algorithm to minimize Dynamic Regret

Let $\{\hat{f}_t\}_{t \geq 1}$, be a sequence of convex cost functions and $\{x_t^\star\}_{t \geq 1}$ any feasible comparator sequence such that $g_t(x_t^\star) \leq 0, \forall t \geq 1$. In other words, the comparators satisfy the online constraint functions at every round. Then the dynamic regret against the comparator sequence $\{x_t^\star\}_{t=1}^{T}$ of any

---

Algorithm 3: Anytime Online Policy for Dynamic COCO

1: **Input:** 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 $\mathcal{X}$

2: **Parameters:** $\alpha = (2GD)^{-1}$.

3: **Initialization:** Set $x_1 = 0, Q(0) = 0$.

4: **for** $t = 1 : T$ **do**

5:     Choose x_t, observe f_t, g_t, incur a cost of $f_t(x_t)$ and constraint violation of $(g_t(x_t))^+$

6:     $\tilde{f}_t \leftarrow \alpha f_t, \tilde{g}_t \leftarrow \alpha \max(0, g_t)$.

7:     $Q(t) = \frac{\lambda_{t-1}}{\lambda_t}Q(t-1) + \tilde{g}_t(x_t)$.

8:     Compute $\hat{f}_t$ as per (9)

9:     Pass $\hat{f}_t$ and $\eta_t = \frac{(D+1)(1+\mathcal{P}_t)}{\sqrt{2\sum_{\tau=1}^{t}(1+\mathcal{P}_\tau)||\nabla_\tau||^2}}$ to Algorithm 1.

10: **end for**

---

online policy which produces an action sequence $\{x_t\}_{t \geq 1}$ is defined to be:

$$\text{D-Regret}_T \equiv \sum_{t=1}^{T} \hat{f}_t(x_t) - \sum_{t=1}^{T} \hat{f}_t(x_t^\star). \quad (16)$$

We define the path length $\mathcal{P}_T$ of a comparator sequence $\{x_t^\star\}_{t=1}^{T}$ as follows:

$$\mathcal{P}_T \equiv \sum_{t=1}^{T-1} ||x_{t+1}^\star - x_t^\star||_2. \quad (17)$$

Note that for the static regret metric defined in (2), we have $x_t^\star = x^\star, \forall t \geq 1$, and hence, the path-length is zero (Hazan 2022). For any round $t \geq 1$, let $x_t^\star$ be an optimal feasible action, *i.e.*, it is a solution to the following constrained convex optimization problem:

$$\min f_t(x), \quad \text{s.t. } g_t(x) \leq 0, \ x \in \mathcal{X}. \quad (18)$$

By definition, the worst-case dynamic regret upper bounds the dynamic regret for any arbitrary feasible comparator sequence. Also, the advantage of considering the *worst-case* benchmarks is that at the end of each round t, we can compute the benchmark $x_t^\star$ by solving the problem 18. This enables us to determine the running value of $\mathcal{P}_t$ at the end of each round t.

For the time being, let us now consider the standard OCO problem without any constraints. The following theorem gives an upper bound to the dynamic regret of the adaptive OGD policy described in Algorithm 1 for the standard OCO problem.

Theorem 5.1 (Dynamic Regret Bounds for AdaGrad)**.** *The AdaGrad policy, described in Algorithm 1, achieves the following adaptive dynamic regret bound for the standard OCO problem against any comparator sequence $u_{1:T}$ whose path length at time t is known to be $\mathcal{P}_t$, using an adaptive learning rate schedule $\eta_t = \frac{(D+1)(1+\mathcal{P}_t)}{\sqrt{2\sum_{\tau=1}^{t}(1+\mathcal{P}_\tau)||\nabla_\tau||^2}}, t \geq 1$:*

$$\text{D-REGRET}_T(\mathcal{P}_T) \leq (D+1)\sqrt{2\sum_{t=1}^{T}(1+\mathcal{P}_t)||\nabla_t||^2}. \quad (19)$$

where $\nabla_t \equiv \nabla f_t(x_t), \forall t \geq 1$ and $diam(\mathcal{X}) = D$.

Remark 5.2. Our proof of Theorem 5.1 adapts the static regret analysis of the AdaGrad algorithm from Orabona (2019, Theorem 4.14) to the dynamic regret setting. See Appendix C for the details. Recently, Supantha and Sinha (2025, Theorem 17) also derived a dynamic regret bound for AdaGrad using similar techniques. The difference between our bound and theirs is that their bound assumes that the total path length $\mathcal{P}_T$ is known in advance, while we only require knowledge of the path length $\mathcal{P}_t$ up to time t. This makes our approach more practical for settings where the full comparator sequence is not known beforehand. Further, our bound is tighter as it incorporates the time-varying path length $\mathcal{P}_t$ inside the summation, whereas their bound uses the total path length $\mathcal{P}_T$ as a multiplicative factor outside the sum. Since $\mathcal{P}_t \leq \mathcal{P}_T$ for all t, the sum $\sum_{t=1}^{T}(1 + \mathcal{P}_t)\|\nabla_t\|^2$ is smaller than $(1 + \mathcal{P}_T)\sum_{t=1}^{T} \|\nabla_t\|^2$, resulting in a tighter overall guarantee. As we shall see, this modified base OCO algorithm would be pivotal for making our adaptive COCO algorithm.

5.2 Dynamic Regret Decomposition Inequality

We use the same surrogate loss as defined earlier in Eqn. (9). This yields

$$\Phi_\tau(Q(\tau)) - \Phi_{\tau-1}(Q(\tau-1)) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x_\tau^\star)$$

$$\stackrel{(a)}{\leq} \Phi'_\tau(Q(\tau))\big(\tilde{g}_\tau(x_\tau) - \tilde{g}_\tau(x_\tau^\star)\big) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x_\tau^\star)$$

$$\stackrel{(b)}{=} \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x_\tau^\star)$$

where in step (a), we have used Eqn. (8) and the feasibility of the benchmark $x_\tau^\star$ (which implies $g_\tau(x_\tau^\star) \leq 0$), and in step (b) we have used the definition of surrogate costs from Eqn. (9). Summing up the above inequalities for $\tau = 1$ to $\tau = t$, we have the following dynamic regret decomposition inequality

$$\Phi_t(Q(t)) + \text{D-Regret}_t(\mathcal{P}_t) \leq \text{D-Regret}'_t(\mathcal{P}_t), \quad (20)$$

where $\text{D-Regret}_t(\mathcal{P}_t)$ and $\text{D-Regret}'_t(\mathcal{P}_t)$ correspond to the Dynamic regrets up to round t for the original cost functions $\{\tilde{f}_\tau\}_{\tau \geq 1}$ and surrogate cost functions $\{\hat{f}_\tau\}_{\tau \geq 1}$ respectively against a sequence of benchmarks with path length at most $\mathcal{P}_t$. Plugging in the dynamic regret expression for the surrogate costs from Theorem 5.1, we conclude that:

$$\Phi_t(Q(t)) + \text{D-Regret}_t(\mathcal{P}_t)$$

$$\leq \sqrt{1 + \mathcal{P}_t}\sqrt{t} + \sqrt{\sum_{\tau=1}^{t}(1 + \mathcal{P}_\tau)\big(\Phi'_\tau(Q(\tau))\big)^2}. \quad (21)$$

The rest of the analysis is similar to that given in Section 4.3 by taking $\Phi_\tau(x) = e^{\lambda_\tau x} - 1$. We now choose $\lambda_\tau = \frac{1}{4\sqrt{\tau(1+\mathcal{P}_\tau)}\sqrt{\log(\tau)+1}\,(\log(\log(\tau)+1)+1)}$. We formally state our results in the following theorem,

Theorem 5.3. *For the COCO problem with adversarially chosen G-Lipschitz cost and constraint functions, Algorithm*

3 yields the following worst-case dynamic regret and CCV bounds for any $t \geq 1$

$$\text{D-Regret}_t(\mathcal{P}_t) \leq 2GD\sqrt{1 + \mathcal{P}_t}(\sqrt{t} + 1),$$

$$CCV_t$$

$$\leq 8GD\sqrt{1 + \mathcal{P}_t}\sqrt{t}\sqrt{\log t + 1}\,\big(\log(\log t + 1) + 1\big)\log 4t,$$

where $\mathcal{P}_t$ is the path-length of the worst-case comparator sequence $\{x_t^\star\}_{\tau=1}^{t}$ up to time t.

6 An Adaptive Optimistic Algorithm

Generalizing the seminal work of Sinha and Vaze (2024), Lekeufack and Jordan (2025) presents an optimistic meta-algorithm for Constrained Online Convex Optimization (COCO) that leverages predictions of the loss and constraint functions to achieve superior performance when those predictions are accurate. However, their algorithm requires tuning a crucial parameter that depends on the total prediction error over the entire horizon - a quantity that is unknown in advance. Consequently, to make their algorithm adaptive, they resort to using the doubling trick. In this section, we propose a modified algorithm that is both optimistic and continuously adaptive, obviating the use of the doubling trick.

6.1 Optimistic meta-algorithm

Algorithm 4: Optimistic COCO

1: **Input:** 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 $\mathcal{X}$
2: **Initialization:** Set $x_1 = 0, Q(0) = 0, Q(1) = 0$.
3: **for** round $t = 1 \ldots T$ **do**
4:     Choose x_t, observe f_t, g_t, incur a cost of $f_t(x_t)$ and constraint violation of $(g_t(x_t))^+$
5:     $\tilde{f}_t \leftarrow \alpha f_t, \tilde{g}_t \leftarrow \alpha \max(0, g_t)$.
6:     Predict $\bar{f}_{t+1}$ and $\bar{g}_{t+1}$.
7:     Compute $\hat{f}_t$ as per (26).
8:     Update $Q(t+1) = \frac{\lambda_t}{\lambda_{t+1}}Q(t) + \tilde{g}_t(x_t)$.
9:     Compute prediction $\bar{\hat{f}}_{t+1}$ as in (23).
10:    Pass $\hat{f}_t$ and $\bar{\hat{f}}_{t+1}$ to Algorithm 5.
11: **end for**

The above algorithm is a synthesis of the Optimistic COCO meta-algorithm proposed in Lekeufack and Jordan (2025) with our novel methodology of time-varying Lyapunov functions and modified queue recursion. It operates by constructing the true and predicted surrogate cost functions ($\hat{f}_t$ and $\bar{\hat{f}}_{t+1}$) using the adaptive queue $Q(t)$ updated via our multiplicative rule, and then passes these functions to a base optimistic OCO algorithm, specifically Optimistic Online Mirror Descent (Algorithm 5), which is detailed in Appendix D.1. The regret guarantee for this base algorithm is given in Theorem D.1. The Optimistic Online Mirror Descent algorithm was first proposed by Rakhlin and Sridharan (2014)

and we just restate the algorithm and its associated regret guarantee in the Appendix. In the following, we provide a brief overview of the optimistic setting, defining the prediction errors and the predicted surrogate cost function used by our algorithm, before stating the main theorem which guarantees its regret and constraint violation bounds. The complete proof is deferred to Appendix D.2.

In the optimistic setting, we assume that at the end of step t, the learner can make predictions $\bar{f}_{t+1}$ and $\bar{g}_{t+1}$. More precisely, we are interested in predictions of the gradients, and, for any function h, we denote by $\nabla \bar{h}_t$ the prediction of the gradient of h. We denote by $\bar{h}_t$ the function whose gradient is $\nabla \bar{h}_t$. Moreover, we define the following prediction errors

$$\epsilon_t(h) := ||\nabla h_t(x_t) - \nabla \bar{h}_t(x_t)||_*^2,$$
$$\mathcal{E}_t(h) := \sum_{\tau=1}^{t} \epsilon_\tau(h), \tag{22}$$

In the previous algorithms, we formed $\hat{f}_t$ at the end of round t using the revealed cost f_t and constraint g_t. Here, we will also form the predicted surrogate cost $\hat{\bar{f}}_{t+1}$ using the predicted cost $\bar{f}_{t+1}$ and constraint $\bar{g}_{t+1}$. We assume without loss of generality that both the actual and predicted functions have the same Lipchitz constant, which upon pre-processing is $(2D)^{-1}$. This also implies that both $\epsilon_t(\tilde{f})$ and $\epsilon_t(\tilde{g})$ are bounded from above by D^{-2}.

We define the predicted surrogate cost $\hat{\bar{f}}_{t+1}$ as-

$$\hat{\bar{f}}_{t+1} = \bar{f}_{t+1} + \Phi'_{t+1}(Q(t+1))\bar{g}_{t+1} \tag{23}$$

However, note that if we were to retain the usual definition of $Q(t)$, the above equation would imply that we know the value of $\tilde{g}_{t+1}(x_{t+1})$ at the end of round t. This is not possible, therefore, Lekeufack and Jordan (2025) has to use delayed updates, wherein,

$$Q(t+1) = Q(t) + \tilde{g}_t(x_t) \tag{24}$$

To apply our adaptive technique we have to augment this with our multiplicative factor to get,

$$Q(t+1) = \frac{\lambda_t}{\lambda_{t+1}}Q(t) + \tilde{g}_t(x_t) \tag{25}$$

Note that, this means $\text{CCV}_t \leq Q(t+1)$. Further, λ_{t+1} should be such that it is known after round t.

Also note that for parity with the definition of predicted surrogate loss, we need to modify our definition of surrogate loss.

$$\hat{f}_t = \tilde{f}_t + \Phi'_t(Q(t))\tilde{g}_t \tag{26}$$

where the $Q(t)$ now has been defined according to Eqn. (24).

We further choose our λ_τ whose value will be justified in subsequent analysis.

$$\lambda_\tau = \frac{1}{c\sqrt{\gamma_\tau + 1}\sqrt{\log(\gamma_\tau + 1) + 1}(\log(\log(\gamma_\tau + 1) + 1) + 1)} \tag{27}$$

. where

$$\gamma_\tau = \mathcal{E}_{\tau-1}(\tilde{g}) + D^{-2} \tag{28}$$

and

$$c = 20(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta}) \tag{29}$$

We summarize our results in the following theorem.

Theorem 6.1. *For the COCO problem with adversarially chosen G-Lipschitz cost and constraint functions, Algorithm 4, using Algorithm 5 as its base OCO algorithm, yields the following Regret and CCV bounds for any $t \in [T]$ where $T \geq 1$ is the horizon length:*

$$Regret_t \leq O\left(\sqrt{\mathcal{E}_t(\tilde{f})}\right),$$
$$CCV_t \leq \tilde{O}\left(\sqrt{\mathcal{E}_t(\tilde{g})}\right).$$

For completeness we provide the proof of the above theorem in section D.2.

7 Simulations

We demonstrate the practical efficacy of our algorithm via numerical experiments on the Online Shortest Path problem with delay constraints - a constrained version of the classic online shortest path problem. We perform numerical simulations on a semi-synthetic dataset constructed from real-world network measurements to model dynamic network conditions with adversarial costs and constraints. Our proposed anytime algorithm is compared against two key baselines: the fixed-horizon algorithm from Sinha and Vaze (2024), which requires advance knowledge of the total horizon T, and its adaptation to an unknown horizon setting via the standard doubling trick. The results, presented in Figures 1 and 2 in the appendix, empirically validate our theoretical claims. The plots show that our anytime algorithm consistently outperforms the doubling trick baseline in both cumulative regret and constraint violation, while also avoiding the instability caused by its periodic restarts. Notably, our algorithm also achieves superior practical performance compared to the fixed-horizon baseline, demonstrating the advantages of its adaptive parameterization. We have detailed the full experimental setup, dataset construction, and analysis of the results in Appendix E.

8 Conclusion

In this work, we proposed anytime algorithms for online convex optimization with adversarial constraints (COCO) applicable across various problems, including the static regret, dynamic regret, and optimistic settings. Our approach employs time-varying Lyapunov functions coupled with a novel multiplicative factor in the virtual queue recursion, a key technical innovation ensuring essential monotonicity properties. This methodology circumvents the need for impractical techniques like the doubling trick and its associated restarts. The resulting algorithms are more practical and stable, benefits which we confirm through numerical simulations on the constrained online shortest path problem.

9 Acknowledgment

References

Besson, L.; and Kaufmann, E. 2018. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*.

Castiglioni, M.; Celli, A.; Marchesi, A.; Romano, G.; and Gatti, N. 2022. A unifying framework for online optimization with long-term constraints. *Advances in Neural Information Processing Systems*, 35: 33589–33602.

Chen, X.; As, Y.; and Krause, A. 2025. Learning Safety Constraints for Large Language Models. In *Forty-second International Conference on Machine Learning*.

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

Gao, F.; Wang, X.; Fan, Y.; Gao, Z.; and Zhao, R. 2024. Constraints driven safe reinforcement learning for autonomous driving decision-making. *IEEE Access*.

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

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

Immorlica, N.; Sankararaman, K.; Schapire, R.; and Slivkins, A. 2022. Adversarial bandits with knapsacks. *Journal of the ACM*, 69(6): 1–47.

Kwon, J.; and Mertikopoulos, P. 2014. A continuous-time approach to online optimization. *arXiv preprint arXiv:1401.6956*.

Lekeufack, J.; and Jordan, M. I. 2025. An Optimistic Algorithm for Online Convex Optimization with Adversarial Constraints. arXiv:2412.08060.

Luo, H.; and Schapire, R. 2014. Towards minimax online learning with unknown time horizon. In *International Conference on Machine Learning*, 226–234. PMLR.

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

Mannor, S.; Tsitsiklis, J. N.; and Yu, J. Y. 2009. Online Learning with Sample Path Constraints. *Journal of Machine Learning Research*, 10(3).

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

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

Rakhlin, A.; and Sridharan, K. 2014. Online Learning with Predictable Sequences. arXiv:1208.3728.

Sarkar, D.; Chakrabartty, A.; Supantha, S.; Dey, P.; and Sinha, A. 2025. Projection-free Algorithms for Online Convex Optimization with Adversarial Constraints. arXiv:2501.16919.

Sinha, A.; and Vaze, R. 2024. Optimal Algorithms for Online Convex Optimization with Adversarial Constraints. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Staff, R. N. 2015. Ripe atlas: A global internet measurement network. *Internet Protocol Journal*, 18(3): 2–26.

Sun, W.; Dey, D.; and Kapoor, A. 2017. Safety-Aware Algorithms for Adversarial Contextual Bandit. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3280–3288. PMLR.

Supantha, S.; and Sinha, A. 2025. Universal Dynamic Regret and Constraint Violation Bounds for Constrained Online Convex Optimization. arXiv:2510.01867.

Williamson, D. P. 2019. *Network flow algorithms*. Cambridge University Press.

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

Yi, X.; Li, X.; Yang, T.; Xie, L.; Hong, Y.; Chai, T.; and Johansson, K. H. 2023. Distributed Online Convex Optimization with Adversarial Constraints: Reduced Cumulative Constraint Violation Bounds under Slater's Condition. *arXiv preprint arXiv:2306.00149*.

Zhang, Z.; Yang, H.; Cutkosky, A.; and Paschalidis, I. C. 2024. Improving adaptive online learning using refined discretization. In *International Conference on Algorithmic Learning Theory*, 1208–1233. PMLR.

Supplementary Material

A The Doubling Trick

The doubling trick is a standard technique for adapting online algorithms that require a known time horizon T to a setting where the horizon is unknown. The method works by running the algorithm in exponentially increasing phases. At the start of each new phase, the algorithm is restarted, and its internal parameters (which depend on T) are recalculated using the new, doubled phase length as the estimate for T.

However, as noted in several prior works, this technique, while theoretically functional, is often criticized for being "aesthetically inelegant and impractical" (Luo and Schapire 2014; Zhang et al. 2024; Kwon and Mertikopoulos 2014). A major line of critique, articulated by Luo and Schapire (2014), is that the method is "intuitively wasteful, since it repeatedly restarts itself, entirely forgetting all the preceding information". Their work, which proposes an alternative adaptive algorithm based on a "pretend prior distribution" over the horizon, also demonstrated empirically that the doubling trick is "beaten by most of the other algorithms" in practice.

Similarly, Kwon and Mertikopoulos (2014) developed their continuous-time approach specifically to provide a "unified any-time analysis without needing to reboot the algorithm every so often". Their work shows that by using a *time-varying parameter* (e.g., $\eta_t \propto 1/\sqrt{t}$) rather than a fixed one, one can achieve optimal $\mathcal{O}(t^{-1/2})$ regret bounds without resorting to restarts.

More recently, Zhang et al. (2024) also motivated their work by designing an algorithm that "does not employ the impractical doubling trick". They reiterated that restarting "wastes data" and causes "large jumps" in the decision sequence, which can be undesirable, and that it performs "considerably worse" in practice. As a solution, they proposed a "refined discretization argument" from a continuous-time model to preserve adaptivity without restarts.

The consensus from these works is that continuously adaptive algorithms are generally preferable and perform better in practice. Our own experiments in Appendix E (Figures 1 and 2) confirm this finding, showing that our intrinsically adaptive method offers distinctive practical performance gains over the standard doubling trick baseline.

B Supporting Lemmas

Lemma B.1. *For*

$$\lambda_\tau = \frac{1}{4\sqrt{\tau}\,\sqrt{\log(\tau)+1}\,\left(\log(\log(\tau)+1)+1\right)},$$

we have

$$\sum_{\tau=1}^{t} \lambda_\tau^2 < \tfrac{1}{4}, \quad \forall\, t \geq 1.$$

Proof. We observe

$$\lambda_\tau^2 = \frac{1}{16\,\tau\,(\log(\tau)+1)\left(\log(\log(\tau)+1)+1\right)^2}.$$

Define

$$a(x) = \frac{1}{16\,x\,(\log(x)+1)\left(\log(\log(x)+1)+1\right)^2}, \quad x \geq 1.$$

The function $a(x)$ is positive and eventually decreasing, so we can apply the integral test:

$$\sum_{\tau=1}^{t} \lambda_\tau^2 \;\leq\; a(1) + \int_1^t a(x)\,dx.$$

We compute

$$\int_1^t a(x)\,dx = \frac{1}{16}\int_1^t \frac{dx}{x(\log x+1)(\log(\log x+1)+1)^2}.$$

With the substitution $u = \log x + 1$ (so $du = dx/x$), this becomes

$$= \frac{1}{16}\int_1^{\log t+1} \frac{du}{u(\log u+1)^2}.$$

Now let $v = \log u + 1$ so that $dv = du/u$. Then

$$= \frac{1}{16}\int_1^{\log(\log t+1)+1} \frac{dv}{v^2} = \frac{1}{16}\left(1 - \frac{1}{\log(\log t+1)+1}\right).$$

Hence

$$\sum_{\tau=1}^{t} \lambda_\tau^2 \le a(1) + \frac{1}{16}\left(1 - \frac{1}{\log(\log t + 1) + 1}\right).$$

Explicitly,

$$a(1) = \frac{1}{16 \cdot 1 \cdot (\log 1 + 1)(\log(\log 1 + 1) + 1)^2} = \frac{1}{16}.$$

Therefore,

$$\sum_{\tau=1}^{t} \lambda_\tau^2 \le \frac{1}{16} + \frac{1}{16} \cdot 1 = \frac{1}{8} < \frac{1}{4}.$$

The bound holds uniformly for all $t \ge 1$, proving the claim. $\qquad\square$

C Proof of Theorem 5.1

Define $y_{t+1} := x_t - \eta_t \nabla \hat{f}_t(x_t)$, where the non-increasing adaptive step size sequence $\{\eta_t\}_{t \ge 1}$ has been defined in Algorithm 1. For any feasible comparator action $x_t^\star \in \mathcal{X}$, we have

$$||x_{t+1} - x_t^\star||^2 \overset{(a)}{\le} ||y_{t+1} - x_t^\star||^2 = ||x_t - x_t^\star||^2 + \eta_t^2 ||\nabla_t||^2 - 2\eta_t \nabla_t^\top (x_t - x_t^\star),$$

where inequality (a) follows from the non-expansive property of Euclidean projection and the second equality follows from the definition of y_{t+1}. Rearranging the above inequality, we have:

$$2\nabla_t^\top (x_t - x_t^\star) \le \frac{||x_t - x_t^\star||^2 - ||x_{t+1} - x_t^\star||^2}{\eta_t} + \eta_t ||\nabla_t||^2. \tag{30}$$

Using the convexity of the cost functions and summing the above inequalities over $1 \le t \le T$, we obtain:

$$2\sum_{t=1}^{T}\left(\hat{f}_t(x_t) - \hat{f}_t(x_t^\star)\right) \le 2\sum_{t=1}^{T} \nabla_t^\top (x_t - x_t^\star) \le \underbrace{\sum_{t=1}^{T} \frac{||x_t - x_t^\star||^2 - ||x_{t+1} - x_t^\star||^2}{\eta_t}}_{(A)} + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2. \tag{31}$$

Next we simplify term (A) in Eqn. (31). It can be expressed as

$$\frac{||x_1^\star - x_1||^2}{\eta_1} - \frac{||x_T^\star - x_{T+1}||^2}{\eta_T} + \sum_{t=1}^{T-1} \frac{||x_{t+1} - x_{t+1}^\star||^2}{\eta_{t+1}} - \frac{||x_{t+1} - x_t^\star||^2}{\eta_t}$$

$$= \frac{||x_1^\star - x_1||^2}{\eta_1} - \frac{||x_T^\star - x_{T+1}||^2}{\eta_T} + \underbrace{\sum_{t=1}^{T-1} \frac{\eta_t ||x_{t+1} - x_{t+1}^\star||^2 - \eta_{t+1}||x_{t+1} - x_t^\star||^2}{\eta_t \eta_{t+1}}}_{(B)}. \tag{32}$$

We next upper bound term (B) in (32).

$$\sum_{t=1}^{T-1} \frac{||\sqrt{\eta_t} x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star||^2 - ||\sqrt{\eta_{t+1}} x_{t+1} - \sqrt{\eta_{t+1}} x_t^\star||^2}{\eta_t \eta_{t+1}}$$

$$= \sum_{t=1}^{T-1} \frac{\langle (\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star - \sqrt{\eta_{t+1}} x_t^\star, (\sqrt{\eta_t} - \sqrt{\eta_{t+1}}) x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star + \sqrt{\eta_{t+1}} x_t^\star \rangle}{\eta_t \eta_{t+1}}$$

$$\le \sum_{t=1}^{T-1} \frac{||(\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star - \sqrt{\eta_{t+1}} x_t^\star||\, ||(\sqrt{\eta_t} - \sqrt{\eta_{t+1}}) x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star + \sqrt{\eta_{t+1}} x_t^\star||}{\eta_t \eta_{t+1}},$$

where the last step follows from an application of the Cauchy-Schwarz inequality. Note that the first term in the numerator can be bounded as:

$$||(\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) x_{t+1} - \sqrt{\eta_t} x_{t+1}^\star - \sqrt{\eta_{t+1}} x_t^\star|| \le \sqrt{\eta_t} ||x_{t+1} - x_{t+1}^\star|| + \sqrt{\eta_{t+1}} ||x_{t+1} - x_t^\star|| \le (\sqrt{\eta_t} + \sqrt{\eta_{t+1}})D,$$

where we have used the triangle inequality and the feasibility of the algorithm's and benchmark's actions in the last step. Using this, we have

$$(B) \leq D \sum_{t=1}^{T-1} \frac{(\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) \, ||(\sqrt{\eta_t} - \sqrt{\eta_{t+1}})x_{t+1} - \sqrt{\eta_t}x_{t+1}^{\star} + \sqrt{\eta_{t+1}}x_t^{\star}||}{\eta_t \eta_{t+1}}$$

$$= D \sum_{t=1}^{T-1} \frac{(\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) ||(\sqrt{\eta_t} - \sqrt{\eta_{t+1}})(x_{t+1} - x_{t+1}^{\star}) + \sqrt{\eta_{t+1}}(x_t^{\star} - x_{t+1}^{\star})||}{\eta_t \eta_{t+1}}$$

$$\overset{(a)}{\leq} D \sum_{t=1}^{T-1} \frac{(\sqrt{\eta_t} + \sqrt{\eta_{t+1}}) \, [(\sqrt{\eta_t} - \sqrt{\eta_{t+1}})D + \sqrt{\eta_{t+1}}||x_t^{\star} - x_{t+1}^{\star}||]}{\eta_t \eta_{t+1}}$$

$$\overset{(b)}{\leq} D \sum_{t=1}^{T-1} \frac{(\eta_t - \eta_{t+1})D + 2\eta_t ||x_{t+1}^{\star} - x_t^{\star}||}{\eta_t \eta_{t+1}}$$

$$= D^2 \left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right) + 2D \sum_{t=1}^{T-1} \frac{||x_{t+1}^{\star} - x_t^{\star}||}{\eta_{t+1}}$$

$$\overset{(c)}{\leq} D^2 \left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right) + \frac{2D\mathcal{P}_T(x_{1:T}^{\star})}{\eta_T},$$

where in step (a), we have used the triangle inequality combined with the feasibility of the algorithm's and benchmark's actions, and in step (c), we have used the definition of the path length of the comparator. The non-increasing property of the step sizes, *i.e.*, $\eta_t \geq \eta_{t+1}, \forall t \geq 1$ was used in steps (a), (b), and (c). Finally, combining the above bound with Eqns. (31) and (32), we conclude

$$2\left(\sum_{t=1}^{T} \hat{f}_t(x_t) - \hat{f}_t(x_t^{\star})\right)$$

$$\leq \frac{||x_1^{\star} - x_1||^2}{\eta_1} - \frac{||x_T^{\star} - x_{T+1}||^2}{\eta_T} + D^2 \left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right) + \frac{2D\mathcal{P}_T(x_{1:T}^{\star})}{\eta_T} + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2$$

$$\leq \frac{D^2 + 2D\mathcal{P}_T(x_{1:T}^{\star})}{\eta_T} + \sum_{t=1}^{T} \eta_t ||\nabla_t||^2.$$

Hence, the dynamic regret of Algorithm 1 can be upper bounded as

$$\text{D-REGRET}_T \leq \frac{\max(D^2, 2D)(1 + \mathcal{P}_T(x_{1:T}^{\star}))}{2\eta_T} + \underbrace{\frac{1}{2} \sum_{t=1}^{T} \eta_t ||\nabla_t||^2}_{(C)}. \tag{33}$$

The dynamic regret bound in Eqn. (33) holds for Algorithm 1 with any non-increasing step sizes. Assuming the path-length is known to be bounded as $\mathcal{P}_T(x_{1:T}^{\star}) \leq \mathcal{P}_T$, using the specific choice of the step size sequence $\eta_t = \frac{(D+1)(1+\mathcal{P}_t)}{\sqrt{2\sum_{\tau=1}^{t}(1+\mathcal{P}_\tau)||\nabla_\tau||^2}}, t \geq 1$, we can upper bound term (C) as follows

$$\frac{1}{2} \sum_{t=1}^{T} \eta_t ||\nabla_t||^2 = \frac{(D+1)}{2\sqrt{2}} \sum_{t=1}^{T} \frac{(1 + \mathcal{P}_t)||\nabla_t||^2}{\sqrt{2\sum_{\tau=1}^{t}(1+\mathcal{P}_\tau)||\nabla_\tau||^2}}$$

$$\leq \frac{(D+1)}{2\sqrt{2}} \int_0^{\sum_{t=1}^{T}(1+\mathcal{P}_t)||\nabla_t||^2} \frac{dz}{\sqrt{z}}$$

$$= \frac{(D+1)}{\sqrt{2}} \sqrt{\sum_{t=1}^{T}(1 + \mathcal{P}_t)||\nabla_t||^2}.$$

Hence, from (33), the dynamic regret for Algorithm 1 can be bounded as

$$\text{D-REGRET}_T(\mathcal{P}_T) \leq \sqrt{2}(D+1)\sqrt{\sum_{t=1}^{T}(1 + \mathcal{P}_t)||\nabla_t||^2}. \tag{34}$$

D   Deferred technical details related to the optimistic setting

D.1   Base Optimistic OCO algorithm and regret guarantee

Algorithm 5: Optimistic Online Mirror Descent

1: **Input** : Convex decision set $\mathcal{X}$, sequence of convex cost functions $\{\hat{f}_t\}_{t=1}^T$, $\mathrm{diam}(\mathcal{X}) = D$.
2: **Initialize** : $x_1 \in \mathcal{X}$ arbitrarily.
3: **for** round $t = 1 \dots T$ **do**
4:     Play action x_t, receive $\hat{f}_t$. Compute $\nabla_t = \nabla \hat{f}_t(x_t)$.
5:     Compute $\eta_{t+1} = \min\left\{ \frac{\sqrt{\beta B}}{\sqrt{\mathcal{E}_t(\hat{f})} + \sqrt{\mathcal{E}_{t-1}(\hat{f})}}, \frac{\beta}{\max_{1 \leq \tau \leq t+1} \bar{L}_\tau} \right\}$
6:     $\tilde{x}_{t+1} := \arg\min_{x \in \mathcal{X}} \langle \nabla_t, x \rangle + \frac{1}{\eta_t} B^R(x; \tilde{x}_t)$.
7:     Compute $\bar{\tilde{f}}_{t+1} = \nabla \tilde{f}_{t+1}(\tilde{x}_{t+1})$.
8:     $x_{t+1} := \arg\min_{x \in \mathcal{X}} \left\langle \bar{\tilde{f}}_{t+1}, x \right\rangle + \frac{1}{\eta_{t+1}} B^R(x; \tilde{x}_{t+1})$.
9: **end for**

Theorem D.1. *(Lekeufack and Jordan 2025, Theorem 10) Let Algorithm 5 be run on a sequence of cost functions $\{\hat{f}_t\}_{t=1}^T$. Let the corresponding sequence of predicted costs be $\{\tilde{f}_t\}_{t=1}^T$. If $\bar{L}_t$ is the Lipchitz constant of the t-th predicted loss, then upon running Algorithm 5 with the following learning rate schedule*

$$\eta_t = \min\left\{ \frac{\sqrt{\beta B}}{\sqrt{\mathcal{E}_{t-1}(\hat{f})} + \sqrt{\mathcal{E}_{t-2}(\hat{f})}}, \frac{\beta}{\max_{1 \leq \tau \leq t} \bar{L}_\tau} \right\} \text{ we get the following regret bound}$$

$$Regret_t \leq 5\sqrt{\frac{B}{\beta}} \sqrt{\mathcal{E}_t(\hat{f})} + 5\frac{B}{\beta} \max_{1 \leq \tau \leq t} \bar{L}_\tau$$

where β is the strong convexity parameter of the regularizer R defining the Bregman divergence, B is an upper bound on this divergence over the algorithm's execution and $\mathcal{E}(\hat{f})$ has been defined in (22).

Remark D.2. The original theorem statement from Lekeufack and Jordan (2025) assumed that $\bar{L}_t \leq \bar{L}_{t+1}$ and thus instead of $\max_{1 \leq \tau \leq t} \bar{L}_\tau$ it only had $\bar{L}_t$. The assumption naturally held true in their case since they made use time-invariant Lyapunov functions as in Sinha and Vaze (2024) and this implied $\bar{L}_t = (1 + \Phi'(Q(t)))G$. In our setting, since $\bar{L}_t = (1 + \Phi'_t(Q(t)))G$, the Lipchitz constants may not be monotonically non-decreasing and to compensate for that, we replace the occurrence of $\bar{L}_t$ in the theorem statement with $\max_{1 \leq \tau \leq t} \bar{L}_\tau$.

D.2   Proof of Theorem 6.1

The proof follows a similar structure to the proof of Theorem 4.2 but incorporates the delayed queue update and the regret bound for the optimistic OCO algorithm.

Using similar analysis as before,

$$\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)) \leq \Phi'_{\tau+1}(Q(\tau+1))\tilde{g}_\tau(x_\tau)$$

Let $x^* \in \mathcal{X}^*$ be a feasible comparator, so $g_\tau(x^*) \leq 0$ for all τ. Consider the quantity $\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*)$:

$$\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*)$$
$$\leq \Phi'_{\tau+1}(Q(\tau+1))\tilde{g}_\tau(x_\tau) + (x_\tau) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*)$$
$$= (\tilde{f}_\tau(x_\tau) + \Phi'_\tau(Q(\tau))\tilde{g}_\tau(x_\tau)) - (\tilde{f}_\tau(x^*) + \Phi'_\tau(Q(\tau))\tilde{g}_\tau(x^*)) + \Phi'_{\tau+1}(Q(\tau+1))\tilde{g}_\tau(x_\tau) - \Phi'_\tau(Q(\tau))\tilde{g}_\tau(x_\tau)$$
$$= \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^*) + \Phi'_{\tau+1}(Q(\tau+1))\tilde{g}_\tau(x_\tau) - \Phi'_\tau(Q(\tau))\tilde{g}_\tau(x_\tau)$$

Also note that,

$$\Phi'_{\tau+1}(Q(\tau+1))\tilde{g}_\tau(x_\tau) - \Phi'_\tau(Q(\tau))\tilde{g}_\tau(x_\tau) \leq \Phi'_{\tau+1}(Q(\tau+1)) - \Phi'_\tau(Q(\tau))$$
$$= \lambda_{t+1}(\Phi_{\tau+1}(Q(\tau+1)) + 1) - \lambda_t(\Phi_\tau(Q(\tau)) + 1)$$
$$\leq \lambda_1(\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)))$$
$$\leq \frac{1}{4}(\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)))$$

where the first inequality follows because $\tilde{g}_\tau(x_\tau) \leq 1$, the second equality follows due to the form of Φ_t, the third inequality follows due to the decreasing nature of λ_t and the last inequality follows because $\lambda_1 \leq \frac{1}{4}$. Therefore, we have,

$$\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)) + \tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*) \leq \hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^*) + \frac{1}{4}(\Phi_{\tau+1}(Q(\tau+1)) - \Phi_\tau(Q(\tau)))$$

Summing from $\tau = 1$ to t:

$$\Phi_{t+1}(Q(t+1)) - \Phi_1(Q(1)) + \sum_{\tau=1}^{t}(\tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*)) \leq \sum_{\tau=1}^{t}\hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^*) + \frac{1}{4}\Phi_{t+1}(Q(t+1))$$

Since $Q(1) = 0$ and $\Phi_1(0) = 0$, and recognizing the regret terms $\text{Regret}_t(x^*) = \sum_{\tau=1}^{t}(\tilde{f}_\tau(x_\tau) - \tilde{f}_\tau(x^*))$ and $\text{Regret}'_t(x^*) = \sum_{\tau=1}^{t}(\hat{f}_\tau(x_\tau) - \hat{f}_\tau(x^*))$,

$$\frac{3}{4}\Phi_{t+1}(Q(t+1)) + \text{Regret}(x^\star) \leq \text{Regret}'(x^\star) \tag{35}$$

Where $\text{Regret}'_t(x^*)$ is the regret of the base optimistic OCO algorithm (Algorithm 5) run on the sequence of surrogate costs $\{\hat{f}_\tau\}_{\tau=1}^{t}$. We bound $\text{Regret}'_t(x^*)$ using Theorem D.1:

$$\text{Regret}'_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{\mathcal{E}_t(\hat{f})} + 5\frac{B}{\beta}\max_{1 \leq \tau \leq t}\bar{L}_\tau$$

Substituting this back:

$$\frac{3}{4}\Phi_{t+1}(Q(t+1)) + \text{Regret}_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{\mathcal{E}_t(\hat{f})} + 5\frac{B}{\beta}\max_{1 \leq \tau \leq t}\bar{L}_\tau \tag{36}$$

First, we bound the surrogate prediction error $\mathcal{E}_t(\hat{f})$ using Eq. (29):

$$\sqrt{\mathcal{E}_t(\hat{f})} = \sqrt{\sum_{\tau=1}^{t}(\hat{f}_\tau(x_\tau) - \bar{\hat{f}}_\tau(x_\tau))^2} = \sqrt{\sum_{\tau=1}^{t}(\tilde{f}_t + \Phi'_t(Q(t))\tilde{g}_t - \bar{f}_t - \Phi'_t(Q(t))\bar{g}_t)^2}$$

$$\leq \sqrt{\sum_{\tau=1}^{t} 2(\tilde{f}_\tau(x_\tau) - \bar{f}_\tau(x_\tau))^2 + \sum_{\tau=1}^{t} 2\Phi'(Q(\tau))^2(\tilde{g}_\tau(x_\tau) - \bar{g}_\tau(x_\tau))^2} \leq \sqrt{\sum_{\tau=1}^{t} 2\epsilon_\tau(f) + \sum_{\tau=1}^{t} 2\Phi'(Q(\tau))^2\epsilon_\tau(\tilde{g})}$$

$$\leq \sqrt{2\mathcal{E}_t(f)} + (\Phi_t(Q(t)) + 1)\sqrt{\sum_{\tau=1}^{t} 2\lambda_\tau^2\epsilon_\tau(\tilde{g})} \tag{37}$$

$$5\sqrt{\frac{B}{\beta}}\sqrt{\mathcal{E}_t(\hat{f})} \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + (\Phi_t(Q(t)) + 1)5\sqrt{\frac{B}{\beta}}\sqrt{\sum_{\tau=1}^{t} 2\lambda_\tau^2\epsilon_\tau(\tilde{g})}$$

We choose λ_τ adaptively based on the prediction error of the constraints as defined before Theorem 6.1. Recall that,

$$\lambda_\tau = \frac{1}{20(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})\sqrt{\gamma_\tau + 1}\sqrt{\log(\gamma_\tau + 1) + 1}(\log(\log(\gamma_\tau + 1) + 1) + 1)}$$

. Note that, $\epsilon_\tau(\tilde{g}) \leq D^{-2}$ (as pre-processed functions are $(2D)^{-1}$-Lipschitz), we can show that $\mathcal{E}_\tau(\tilde{g}) = \mathcal{E}_{\tau-1}(\tilde{g}) + \epsilon_\tau(\tilde{g}) \leq \mathcal{E}_{\tau-1}(\tilde{g}) + D^{-2} \leq \gamma_\tau$. Thus:

$$\lambda_\tau \leq \frac{1}{20(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})\sqrt{\mathcal{E}_\tau(\tilde{g}) + 1}\sqrt{\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1}(\log(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1) + 1)}$$

$$\implies \lambda_\tau^2 \leq \frac{1}{400(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})^2(\mathcal{E}_\tau(\tilde{g}) + 1)(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1)(\log(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1) + 1)^2}$$

$$\implies 2\epsilon_\tau(\tilde{g})\lambda_\tau^2 \leq \frac{\epsilon_\tau(\tilde{g})}{200(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})^2(\mathcal{E}_\tau(\tilde{g}) + 1)(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1)(\log(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1) + 1)^2}$$

$$\implies 2\epsilon_\tau(\tilde{g})\lambda_\tau^2 \leq \frac{(\mathcal{E}_\tau(\tilde{g}) + 1) - (\mathcal{E}_{\tau-1}(\tilde{g}) + 1)}{200(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})^2(\mathcal{E}_\tau(\tilde{g}) + 1)(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1)(\log(\log(\mathcal{E}_\tau(\tilde{g}) + 1) + 1) + 1)^2}$$

. The above form makes it clear that $\sum_{\tau=1}^{t} 2\lambda_\tau^2 \epsilon_\tau(\tilde{g})$ is amenable to the integral-sum trick as in Lemma B.1. Overall, we get,

$$\sum_{\tau=1}^{t} 2\lambda_\tau^2 \epsilon_\tau(\tilde{g}) \leq \frac{1}{100(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})^2} \implies 5\sqrt{\frac{B}{\beta}}\sqrt{\sum_{\tau=1}^{t} 2\lambda_\tau^2 \epsilon_\tau(\tilde{g})} \leq \frac{1}{2}$$

Overall, we get that,

$$\sqrt{\mathcal{E}_t(\hat{f})} \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{1}{2}\Phi_t(Q(t)) + \frac{1}{2}$$

Then, we bound $\max_{1 \leq \tau \leq t} \bar{L}_\tau$,

$$\max_{1 \leq \tau \leq t} \bar{L}_t \leq (2D)^{-1} \max_{1 \leq \tau \leq t}(1 + \lambda_\tau e^{\lambda_\tau Q(\tau)}) \leq (2D)^{-1} + (2D)^{-1} \max_{1 \leq \tau \leq t} \lambda_\tau e^{\lambda_\tau Q(\tau)} \leq (2D)^{-1} + (2D)^{-1}\lambda_1 e^{\lambda_t Q(t)}$$

Note that, $5\frac{B}{\beta}\lambda_1 \leq \frac{1}{4D^{-1}}$, so $(2D)^{-1}5\frac{B}{\beta}\lambda_1 \leq \frac{1}{8}$. Overall, we get,

$$5\frac{B}{\beta}\max_{1 \leq \tau \leq t} \bar{L}_t \leq 5\frac{B}{\beta}(2D)^{-1} + \frac{1}{8}(\Phi_t(Q(t)) + 1)$$

Substituting the upper-bound of $5\sqrt{\frac{B}{\beta}}\sqrt{\mathcal{E}_t(\hat{f})}$ and $5\frac{B}{\beta}\max_{1 \leq \tau \leq t} \bar{L}_\tau$ in inequality (36):

$$\frac{3}{4}\Phi_{t+1}(Q(t+1)) + \text{Regret}_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{1}{2}\Phi_t(Q(t)) + \frac{1}{2} + 5\frac{B}{\beta}(2D)^{-1} + \frac{1}{8}(\Phi_t(Q(t)) + 1)$$

Further, note that, $\Phi_t(Q(t)) \leq \Phi_{t+1}(Q(t+1))$,

$$\frac{3}{4}\Phi_{t+1}(Q(t+1)) + \text{Regret}_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{5}{8} + 5\frac{B}{\beta}(2D)^{-1} + \frac{5}{8}\Phi_{t+1}(Q(t+1))$$

$$\frac{1}{8}\Phi_{t+1}(Q(t+1)) + \text{Regret}_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{5}{8} + 5\frac{B}{\beta}(2D)^{-1}$$

Dropping the non-negative $\Phi_t(Q(t))$ term gives the regret bound for the pre-processed functions:

$$\text{Regret}_t(x^*) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{5}{8} + 5\frac{B}{\beta}(2D)^{-1}$$

For the CCV bound, use the trivial lower bound $\text{Regret}_t(x^*) \geq -Dt/(2D) \cdot \alpha = -t/2$. Substitute this into the inequality before dropping the Φ term:

$$\frac{1}{8}\Phi_{t+1}(Q(t+1)) \leq 5\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + \frac{5}{8} + 5\frac{B}{\beta}(2D)^{-1} + \frac{t}{2}$$

$$\Phi_{t+1}(Q(t+1)) \leq 40\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + 10 + 20\frac{B}{\beta D} + 4t$$

Since $\Phi_t(x) = e^{\lambda_t x} - 1$:

$$e^{\lambda_{t+1}Q(t+1)} \leq 40\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + 11 + 20\frac{B}{\beta D} + 4t$$

$$Q(t+1) \leq \frac{1}{\lambda_{t+1}}\log(40\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + 11 + 20\frac{B}{\beta D} + 4t)$$

$$Q(t+1)$$
$$\leq 20(\sqrt{\frac{B}{\beta}} + \frac{B}{\beta})\sqrt{\gamma_{t+1} + 1}\sqrt{\log(\gamma_{t+1} + 1) + 1}(\log(\log(\gamma_{t+1} + 1) + 1) + 1)\log(40\sqrt{\frac{B}{\beta}}\sqrt{2\mathcal{E}_t(f)} + 11 + 20\frac{B}{\beta D} + 4t)$$

where $\gamma_{t+1} = \mathcal{E}_t(\tilde{g}) + D^{-2}$.

Overall, we get a bound of $O(\sqrt{\mathcal{E}_t(\tilde{f})})$ for regret and a bound of $\tilde{O}(\sqrt{\mathcal{E}_t(\tilde{g})})$ for the CCV.

E   Numerical Simulations

Setup: We consider a constrained version of the online shortest path problem (Hazan 2022), where the length corresponds to the latency and the constraint is on the long-term cumulative bandwidth across the path. In this problem, on each round, the online algorithm first selects a route connecting a source s to a destination d on a graph $G(V, E)$. The latency and bandwidth of each edge vary across rounds, reflecting dynamic network conditions. The objective is to minimize the cumulative latency subject to a long-term lower bound on the cumulative bandwidth. Formally, the following sequence of events takes place on the tth round:

1. The algorithm first chooses a route (randomly or otherwise) $p_t \in P_{s,d}$, where $P_{s,d}$ is the set of all $s - d$ routes in the graph.
2. A latency of $\tau_e(t)$ and a bandwidth of $l_e(t)$ is chosen by an adversary for each edge $e \in E$.
3. The algorithm incurs a latency cost of $\sum_{e \in p_t} \tau_e(t)$ and a bandwidth cost of $-\sum_{e \in p_t} l_e(t)$ on round t.

We represent each route p by its corresponding $|E|$-dimensional binary incidence vector where $p_e = 1$ if the edge $e \in E$ belongs to the route or $p_e = 0$ otherwise. On each round, our online policy returns an element from the convex hull of $P_{s,d}$, also known as the unit flow polytope (Hazan 2022). We use Dijkstra's algorithm for computing the weighted shortest route. It is well-known that any element in the unit flow polytope can be efficiently decomposed into a convex combination of at most $|E|$ number of $s - d$ routes using the flow decomposition lemma (Williamson 2019, Lemma 2.20). These convex combinations can be used to randomly select a single route on each round, incurring the same expected cost. The experiments are performed on a quad-core CPU with 8 GB RAM. The experimental setup mostly follows that in Sarkar et al. (2025).

Dataset: For the experiments, we utilize a semi-synthetic dataset designed to simulate dynamic network conditions, constructed following methodologies established in prior work (Sarkar et al. 2025) for evaluating online network algorithms. The dataset originates from raw data collected via public probes on the RIPE Atlas global network measurement platform (Staff 2015), specifically HTTPS measurements capturing bandwidth and latency between network nodes.The resulting graph consists of $n = 191$ nodes and $m = 1200$ edges, where each edge is characterized by latency and bandwidth attributes. The construction involves centering the graph around a hub node, adding connections based on the RIPE Atlas data, and then introducing additional random edges, with edge weights determined by latencies relative to the hub and minimum bandwidths between connected nodes.To mimic real-world fluctuations for the online setting, temporal variations are introduced through random scaling factors applied at each iteration: latency values are scaled randomly between 0.5x and 1.5x, and bandwidth values between 0.8x and 1.2x. This process generates the time-varying latency and bandwidth matrices used for evaluating algorithm performance over a horizon length of $T = 1600$ iterations. This dataset provides a realistic and challenging testbed for constrained online optimization algorithms.
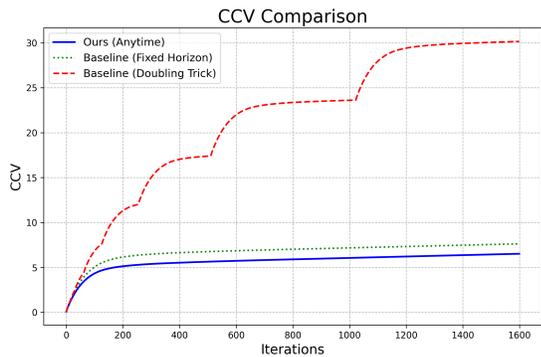

Figure 1: CCV comparison between our policy, Algorithm 1 of Sinha and Vaze (2024) and its doubling trick implementation.
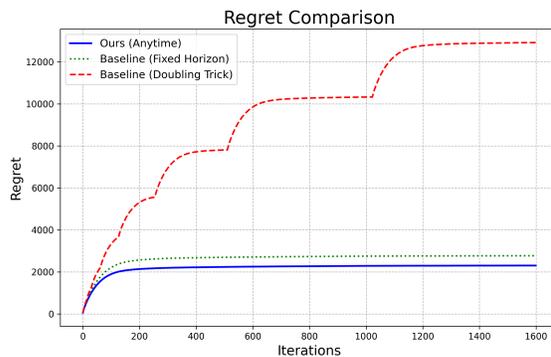

Figure 2: Regret comparison between our policy, Algorithm 1 of Sinha and Vaze (2024) and its doubling trick implementation..

Results: The empirical results presented in Figure 1 and Figure 2 provide strong validation for our theoretical contributions. The plots compare the performance of our proposed anytime algorithm against two key baselines: the standard fixed-horizon algorithm from Sinha and Vaze (2024), which assumes prior knowledge of the total horizon T, and its adaptation to an unknown horizon setting via the standard doubling trick.

In summary, the simulations reveal a key practical advantage of our approach, demonstrating that our anytime algorithm outperforms both the fixed-horizon and doubling trick baselines.

The doubling trick based method performs the worst. This is due to the restarting technique that they employ where an the algorithm has to discard the information it has upto that point and start from scratch. Our observations match those in

(Besson and Kaufmann 2018) where they evaluated several versions of the doubling trick and concluded that performance is substantially worse than that of an intrinsically anytime algorithm, even one with less favorable theoretical guarantees. Moreover, this performance gap becomes larger each time the algorithm is restarted.

While the fixed-horizon method is optimal in a worst-case theoretical sense, its use of a single, conservative trade-off parameter hinders its practical performance. Our algorithm, with its time-varying parameter, is more adaptive; it aggressively penalizes violations in the early stages, quickly learning the constraints and achieving a superior performance trajectory. This confirms that our method is not just a tool for handling an unknown horizon, but a fundamentally more effective strategy that also avoids the instability and performance loss inherent in the doubling trick.