Safe Online Convex Optimization with Unknown Linear Safety Constraints

Sapana Chaudhary [1], **Dileep Kalathil** [1]

[1] Department of Electrical and Computer Engineering, Texas A&M University, USA
sapanac@tamu.edu, dileep.kalathil@tamu.edu

Abstract

We study the problem of safe online convex optimization, where the action at each time step must satisfy a set of linear safety constraints. The goal is to select a sequence of actions to minimize the regret without violating the safety constraints at any time step (with high probability). The parameters that specify the linear safety constraints are unknown to the algorithm. The algorithm has access to only the noisy observations of constraints for the chosen actions. We propose an algorithm, called the Safe Online Projected Gradient Descent (SO-PGD) algorithm, to address this problem. We show that, under the assumption of the availability of a safe baseline action, the SO-PGD algorithm achieves a regret $O(T^{2/3})$. While there are many algorithms for online convex optimization (OCO) problems with safety constraints available in the literature, they allow constraint violations during learning/optimization, and the focus has been on characterizing the cumulative constraint violations. To the best of our knowledge, ours is the first work that provides an algorithm with provable guarantees on the regret, without violating the linear safety constraints (with high probability) at any time step.

1 Introduction

Online learning/optimization is a sequential decision making paradigm, where the decision maker adaptively selects a sequence of actions based on the past observations (Cesa-Bianchi and Lugosi 2006). Online convex optimization (OCO) is an important class of online optimization problems, where the cost function faced by the decision maker at each time step is an arbitrarily-varying convex function (Hazan 2016; Shalev-Shwartz 2011). In the OCO problem, a sequence of arbitrarily-varying convex cost functions $\{f_t, t = 1, \ldots, T\}$ are revealed, one per time step, to the decision maker. The decision maker selects an action x_t from a convex set $\mathcal{X}$, *before* the cost function f_t is revealed. The typical performance objective is to minimize the regret, which characterizes the difference between cumulative cost incurred by the decision maker and that of an oracle algorithm that employs the best fixed action in hindsidght at all time steps. There are a number of OCO algorithms that achieve different sublinear regret guarantees with different computational complexity (Hazan 2016).

In many real-world applications, however, the actions selected by the decision maker must satisfy some necessary safety constraints over the set $\mathcal{X}$. For example, in power systems, the control actions that decide the demand management should not violate the line flow and the voltage regulation constraints (Dobbe et al. 2020). In communication networks, the transmission rate is limited by constraints on the maximum allowable radiated power due to interference and human safety considerations (Luong et al. 2019). In robotics applications, the control actions should maintain the closed-loop stability of the system (Åström and Murray 2010). Typically, such constraints are represented as a *safe set* $\mathcal{X}^s$ and the control action x_t must lie inside $\mathcal{X}^s$ *for all* t for the safe operation of the system.

Often, the safe set $\mathcal{X}^s$ is determined by the parameters of the system that are typically *unknown* to the decision maker a-priori. For example, in power systems, the constraints on the control actions depend on the line parameters, which are typically unknown. In robotics, designing a closed-loop stable controller requires the dynamic model of the robot, which may be unknown. Thus, the decision maker has to learn the unknown parameters to characterize the unknown safe set. While an exploration algorithm can be used to estimate these parameters, such algorithms often take random actions for efficient estimation that may violate the safety constraints. Moreover, taking actions with respect to an estimated safe set may still violate the safety constraints due to the unavoidable estimation errors.

In this paper, we address the problem of *safe online convex optimization* with an *unknown safe set*, where the decision maker has access to only noisy observations of the safety constraints (that define the safe set) for the chosen actions. Our goal is to design an algorithm that minimizes the regret *while satisfying the safety constraints at all time steps*.

While there are many works in the OCO literature that address the problem with safety constraints (see the related works section below), they typically allow constraint violations during learning/optimization. The main goal of such algorithms is then to obtain a (sublinear) bound on the cumulative constraint violations, in addition to the standard regret. In sharp contrast to such works, we focus on designing an algorithm that satisfies the safety constraints *at all time steps* while providing a provable guarantee on the regret.

In this paper, we restrict ourselves to the setting where

the unknown safe set is a closed polytope characterized by a set of linear inequalities with unknown parameters. We believe that addressing the linear constraint setting is a natural first step towards developing a fundamental understanding of safe OCO algorithms for general non-linear setting. To the best of our knowledge, this is the first work that addresses the safe OCO problem with a provable guarantee on satisfying the safety constraints at all time steps, even in a setting with linear constraints.

1.1 Related Work

OCO: The OCO problem was first formally addressed in (Zinkevich 2003), though some prior works (Cesa-Bianchi, Long, and Warmuth 1996; Gordon 1999) had considered similar settings. In (Zinkevich 2003), the author proposed an online gradient descent algorithm and showed that it achieves $O(\sqrt{T})$ regret. A number of OCO algorithms under different assumptions have been developed since, see the monographs (Shalev-Shwartz 2011; Hazan 2016).

OCO with Long Term Constraints: Most of the standard OCO algorithms assume full knowledge of the constraint set $\mathcal{X}^s$. However, in many real-world applications, the constraint set $\mathcal{X}^s$ is often specified in terms of the functional inequalities, i.e., $\mathcal{X}^s = \{x \in \mathcal{X} : g_{i,t}(x) \leq 0, i \in \{1, \ldots, m\}, t \in \{1, \ldots, T\}\}$, where $g_{i,t}$s are convex functions. The OCO with long term constraints problem considers a relaxed version of such constraints, where the goal is to bound the constraint violations, $\max_i \sum_{t=1}^{T} g_{i,t}(x_t)$, instead of satisfying the constraints at each time step.

The OCO with long term constraint problem was first introduced in (Mahdavi, Jin, and Yang 2012), which assumed that the constraint functions are the same for all t, i.e., $g_{i,t} = g_i, \forall t$. For deterministic constraints, the algoirthm proposed in (Mahdavi, Jin, and Yang 2012) achieves $O(T^{1/2})$ regret and $O(T^{3/4})$ constraint violation. Recently, (Yu and Neely 2020) showed that it is possible to achieve $O(T^{1/2})$ regret and $O(1)$ constraint violation. In (Yu, Neely, and Wei 2017), the authors addressed the stochastic constraints setting, where the constraint functions are of the form $g_{i,t}(x) = g_i(x, \omega_t)$, where w_ts are i.i.d. random variables. They proposed an algorithm that simultaneously achieves $O(\sqrt{T})$ regret and (expected) constraint violation. A recent work (Wei, Yu, and Neely 2020) has improved this result by removing some assumptions while maintaining the regret guarantees.

In (Neely and Yu 2017), the authors addressed the setting where the constraint functions $g_{i,t}$s are arbitrarily-varying (adversarial), and proposed an algorithm with $O(T^{1/2})$ regret and constraint violation. This problem was also addressed in (Sun, Dey, and Kapoor 2017; Chen, Ling, and Giannakis 2017; Cao and Liu 2018). A distributed version of this problem has been studied recently in (Yi et al. 2020).

We emphasize that all the above mentioned works allow constraint violations during learning/optimization. Significantly different from these, we propose an algorithm that does not violate the unknown linear constraints that define the safe set at any time step during learning/optimization.

Safe Learning/Optimization: The works closest to our setting are (Amani, Alizadeh, and Thrampoulidis 2019) and (Khezeli and Bitar 2020), where the authors addressed the linear bandits problem with unknown linear safety constraints that have to be satisfied at all time steps during learning. For ensuring safe exploration in the initial phase of learning, they introduce an assumption about the availability of a known safe baseline action. They showed that $O(T^{1/2})$ regret is achievable without safety constraints violations during learning if a lower bound on the distance between the optimal action and the boundary of the safe set is known. If such a lower bound is not available, then $O(T^{2/3})$ regret is achievable. Instead of the static linear cost function considered in these works, we consider the more challenging arbitrarily-varying convex cost functions. Moreover, we also consider a set of linear constraints as opposed to a single linear constraint studied in these works.

Convex optimization with unknown linear safety constraints addressed in (Usmanova, Krause, and Kamgarpour 2019) and (Fereydounian et al. 2020) is another class of works that is close to ours. Similar to (Amani, Alizadeh, and Thrampoulidis 2019; Khezeli and Bitar 2020) these works also make use of the assumption of a safe baseline action. They consider a static convex cost function and focus on characterizing the sample complexity, which is quite different from our setting (arbitrarily-varying cost functions) and objective (regret minimization).

1.2 Main Contributions

We formulate the safe online convex optimization problem where the action must satisfy a set of *unknown* linear safety constraints *at all time steps*. The decision maker has only access to a noisy measurement of the constraints with respect to the chosen action at each time step. We propose a new algorithm, called the Safe Online Projected Gradient Descent (SO-PGD) algorithm, and show that this algorithm achieves $O(T^{2/3})$ regret while satisfying the safety constraints at all time steps, with a high probability. To the best of our knowledge, this is the first such result in the OCO literature, even in a setting with liner constraints.

Similar to (Amani, Alizadeh, and Thrampoulidis 2019; Khezeli and Bitar 2020; Usmanova, Krause, and Kamgarpour 2019; Fereydounian et al. 2020), our algorithm also makes use of the assumption of a safe baseline action for initial exploration and for estimating the unknown parameters. However, a naive estimate of the safe set may lead to constraint violations because of the inherent estimation error. The key idea we use is the construction of a conservative safe set that is provably a subset of the unknown safe set. Our algorithm performs online gradient descent with respect to this conservative safe set, which provably ensures that safety constraints are satisfied at each time step. We then characterize the error because of using this conservative safe set. We show that a clever balancing of the exploration and online optimization can achieve $O(T^{2/3})$ regret without constraint violations at any time steps.

1.3 Notations

For any positive semidefinite matrix A, we denote $\|x\|_A = \sqrt{x^\top Ax}$. For any square matrix A, we denote its minimum and maximum eigenvalues by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, respectively. For any two integer M_1, M_2 with $M_1 < M_2$, we denote $[M_1, M_2] = \{M_1, M_1+1, \ldots, M_2\}$. For any random vector ζ, $\text{Cov}(\zeta) = \mathbb{E}[\zeta\zeta^\top]$. For any convex set $\mathcal{X} \subset \mathbb{R}^n$ and any $x \in \mathbb{R}^n$, $\Pi_{\mathcal{X}}(x)$ denotes the projection of x to $\mathcal{X}$ with respect to the Euclidean norm.

Due to page limit, we omit the detailed proofs. All the proofs are available in the online supplement (Chaudhary and Kalathil 2021).

2 Safe Online Convex Optimization: Problem Formulation

The general framework of online convex optimization (Hazan 2016) is as follows: at each time step t, the algorithm selects an action $x_t \in \mathcal{X} \subset \mathbb{R}^d$ and incurs a cost $f_t(x_t)$, where $f_t : \mathcal{X} \to \mathbb{R}$ is a convex function. The cost function f_t is not known at the time of making the decision x_t, and the sequence of cost functions $\{f_t, t \in [1,T]\}$ is assumed to be arbitrary. In addition to the incurred cost $f_t(x_t)$, it is generally assumed that the value of the gradient of f_t evaluated at x_t, $\nabla f_t(x_t)$, is also available to the algorithm. The goal of a standard online convex optimization algorithm is to select a sequence of actions $\{x_t, t \in [1,T]\}$ in order to minimize the regret defined as $\sum_{t=1}^T f_t(x_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^T f_t(x)$. Most of the existing works assume that the set $\mathcal{X}$ is known to the algorithm a priori.

In this work, we consider the safe online convex optimization problem with an unknown safe set characterized by a set of unknown linear safety constraints. More precisely, at each time step t, the algorithm has to take an action x_t from the safe set $\mathcal{X}^s$, defined as

$$\mathcal{X}^s = \{x \in \mathbb{R}^d : Ax \leq b\}, \tag{1}$$

where the matrix $A \in \mathbb{R}^{m\times d}$ and the vector $b \in \mathbb{R}^m$. Denoting $A = [a_1, a_2, \ldots, a_m]^\top, b = [b_1, b_2, \ldots, b_m]^\top$, where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}^1$, the safe set $\mathcal{X}^s$ is defined in terms of m linear constraints, and the ith linear constraint is of the form $a_i^\top x \leq b_i$. Thus, $\mathcal{X}^s$ is closed polytope. The matrix A is unknown to the algorithm a priori. So, the safe set $\mathcal{X}^s$ is also unknown. For simplifying the analysis, we assume that b is known to the algorithm.

It is impossible to learn the safety constraints if the algorithm receives no information that can be used to estimate the unknown safe set $\mathcal{X}^s$, or equivalently, the unknown parameter A. Here, we make a natural assumption that the algorithm receives a noisy observation $y_t \in \mathbb{R}^m$ at each time step t, where $y_t = Ax_t + w_t$, and w_t is a zero mean sub-Gaussian noise.

The goal of the safe online convex optimization algorithm is to select a sequence of actions $\{x_t, t \in [1,T]\}$ in order to minimize the regret $R(T)$, defined as

$$R(T) = \sum_{t=1}^T f_t(x_t) - \min_{x\in\mathcal{X}^s} \sum_{t=1}^T f_t(x), \tag{2}$$

while simultaneously satisfying the safety constraints by ensuring that

$$\mathbb{P}(x_t \in \mathcal{X}^s, \text{for all } t \in [1,T]) \geq 1 - \delta, \tag{3}$$

for a given $\delta \in (0,1)$.

2.1 Model Assumptions

In order to analyze the safe OCO problem stated above, we make the following assumptions.

Assumption 1 (Cost Functions)**.** *The cost functions $\{f_t, t \in [1,T]\}$ are convex and have a bounded gradient, i.e., $\max_{t\in[1,T]} \max_{x\in\mathcal{X}} \|\nabla f_t(x)\| \leq G$.*

The above assumption is standard in the OCO literature. Also, this assumption implies that f_ts are G-Lipschitz.

Assumption 2 (Boundedness)**.** *(i) $\|x\|_2 \leq L, \forall x \in \mathcal{X}^s$.*
(ii) $\max_{i\in[1,m]} \|a_i\|_2 \leq L_A$.

These are also standard assumptions in the linear bandits and OCO literature. Also, as is standard in the literature, we assume that G, L, L_A are known to the algorithm.

Assumption 3 (Sub-Gaussian Noise)**.** *The noise sequence $\{w_t, t \in [1,T]\}$ is R-sub-Gaussian with respect to a filtration $\{\mathcal{F}_t, t \in [1,T]\}$, i.e.,*
(i) $\mathbb{E}[w_t|\mathcal{F}_{t-1}] = 0, \forall t, t \in [1,T]$,
(ii) $\mathbb{E}[e^{\lambda w_t} \mid \mathcal{F}_{t-1}] \leq \exp(\lambda^2 R^2/2), \forall \lambda \in \mathbb{R}, \forall t \in [1,T]$.

Since the safe set $\mathcal{X}^s$ is unknown, clearly it is not possible to satisfy safety constraints right from the first time step without making any additional assumptions. We overcome this obvious limitation by assuming that the algorithm has access to a safe baseline action x^s such that $x^s \in \mathcal{X}^s$. We formalize this assumption as follows.

Assumption 4 (Safe Baseline Action)**.** *There exists a safe baseline action $x^s \in \mathcal{X}^s$ such that $Ax^s = b^s < b$. The algorithm knows x^s and b^s and hence the safety gap $\Delta^s = \min_i(b_i - b_i^s)$.*

This assumption is similar to that of the safe baseline action assumption used in the context of safe linear bandits and safe convex optimization (Amani, Alizadeh, and Thrampoulidis 2019; Khezeli and Bitar 2020; Usmanova, Krause, and Kamgarpour 2019; Fereydounian et al. 2020). The key intuition is that, any algorithm used in a real-world decision making problem has to perform at least as well as a baseline action, which is often conservatively designed to satisfy the safety constraints. Typically, this baseline action is already employed to solve the real-world decision making problem and there will be large amount of data generated according to this baseline action, which can be used to estimate the value b^s. We emphasize that while the baseline action is safe by definition, it may be far way from the optimal action that minimizes the regret.

3 Safe Online Projected Gradient Descent (SO-PGD) Algorithm

We propose an algorithm, which we call the *safe online projected gradient descent* (SO-PGD) algorithm, to solve the online convex optimization problem with unknown linear

safety constraints. The SO-PGD Algorithm is formally given in Algorithm 1. It has three main parts: (i) safe exploration, (ii) conservative safe set estimation, and (iii) online gradient descent.

Algorithm 1: SO-PGD Algorithm

Input: $\gamma, \eta, T_0, \delta, x^s, T$
Safe exploration:
for $t = 1, \ldots, T_0$ **do**
    Select action $x_t = (1-\gamma)x^s + \gamma\zeta_t$ (as in (4))
end for
Conservative safe set estimation:
Estimate $\hat{A}$ according to (5)
Compute conservative safe set $\hat{\mathcal{X}}^s$ according to (8)
Online gradient descent:
for $t = T_0 + 1, \ldots, T$ **do**
    $x_{t+1} = \Pi_{\hat{\mathcal{X}}^s}(x_t - \eta\nabla f_t(x_t))$
end for

3.1 Safe Exploration

The goal of the safe exploration part of the SO-PGD algorithm is to estimate the safe set without violating the safety constraints. This is achieved by pursuing a pure exploration strategy for the first T_0 time steps by carefully selected exploration actions. Since the safety constraints have to be satisfied at all time steps, we make use of the knowledge of the safe baseline action x^s to collect the observations that are necessary for estimating the safe set. However, since $y^s = Ax^s$ may not be a function of all the elements of A, taking the safe baseline action x^s alone will not give a good estimate of the unknown parameter A. To overcome this issue, we design exploration actions as random perturbation around x^s in such a way that they do not violate the safety constraints. More formally, for any time step $t \in [1, T_0]$, the safe exploration action x_t is selected as

$$x_t = (1-\gamma)x^s + \gamma\zeta_t, \tag{4}$$

for some $\gamma \in [0, 1)$, where ζ_ts are i.i.d. zero mean random vectors such that $\|\zeta_t\| \leq \min\{1, L\}$ and $\text{Cov}(\zeta_t) = \sigma_\zeta^2 I$ for all t. By controlling the value of γ, we can ensure that the exploration action x_t satisfies the safety constraints for all $t \in [1, T_0]$, as shown below.

Lemma 1. *Let Assumption 2 and 4 hold. Let* $\gamma = \frac{\Delta^s}{L_A}$. *Then, the safe exploration action* x_t *given in* (4) *satisfies the safety constraints* $Ax_t \leq b$ *for all* $t \in [1, T_0]$ *almost surely.*

3.2 Estimation of Conservative Safe Set

At the end of the safe exploration phase, using the past exploration actions x_t and the past observations $y_t = Ax_t + w_t$, $t \in [1, T_0]$, the algorithm computes the ℓ_2-regularized least squares estimate $\hat{A}$ of the matrix A. More formally, let $X_{T_0} = [x_1, \ldots, x_{T_0}]^\top \in \mathbb{R}^{T_0 \times d}$ and $Y_{T_0} = [y_1, \ldots, y_{T_0}]^\top \in \mathbb{R}^{T_0}$. Then, the ℓ_2-regularized least squares estimate is given by

$$\hat{A} = (\lambda I + X_{T_0}^\top X_{T_0})^{-1} X_{T_0}^\top Y_{T_0}. \tag{5}$$

We denote $\hat{A} = [\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_m]^\top$, where $\hat{a}_i$ is the estimate of a_i.

The SO-PGD algorithm next constructs the ellipsoidal confidence set $\mathcal{C}_i(\delta)$ around $\hat{a}_i, i \in [1, m]$, that contains the unknown parameter a_i with a probability greater than $(1 - \delta/m)$. More formally, we define

$$\mathcal{C}_i(\delta) = \{a \in \mathbb{R}^d : \|a - \hat{a}_i\|_{V_{T_0}} \leq \beta_{T_0}(\delta)\}, \tag{6}$$

where V_{T_0} is the Gram matrix of the least squares estimation, given by $V_{T_0} = \lambda I + X_{T_0}^\top X_{T_0} = \lambda I + \sum_{t=1}^{T_0} x_t x_t^\top$, and

$$\beta_{T_0}(\delta) = R\sqrt{d\log\left(\frac{1 + T_0 L^2/\lambda}{\delta/m}\right)} + \sqrt{\lambda}L_A. \tag{7}$$

The radius $\beta_{T_0}(\delta)$ of the confidence of set $\mathcal{C}_i(\delta)$ is selected in order to to ensure that the true parameter a_i is inside it with high probability. We note that this is a standard approach used in the linear bandits literature (Abbasi-Yadkori, Pál, and Szepesvári 2011, Theorem 2). We formally state this result below.

Lemma 2. *Let Assumption 2 and 3 hold. Then,* $\mathbb{P}(a_i \in \mathcal{C}_i(\delta), \forall i \in [1, m]) \geq 1 - \delta$.

Now, using the confidence sets $\mathcal{C}_i(\delta), i \in [1, m]$, the algorithm constructs a conservative safe set $\hat{\mathcal{X}}^s$ as

$$\hat{\mathcal{X}}^s = \{x \in \mathbb{R}^d : \tilde{a}_i^\top x \leq b, \forall \tilde{a}_i \in \mathcal{C}_i(\delta), \forall i \in [1, m]\}. \tag{8}$$

Note that the elements of $\hat{\mathcal{X}}^s$ satisfy the safety constraint with respect to *all* elements of the confidence set $\mathcal{C}_i(\delta), \forall i \in [1, m]$. This condition naturally leads to a conservative inner approximation of the true safe set $\mathcal{X}^s$. We formally state this observation below.

Lemma 3. *Let Assumption 2 and 3 hold. Then,* $\hat{\mathcal{X}}^s \subseteq \mathcal{X}^s$ *with probability at least* $1 - \delta$.

Using the conservative safe set $\hat{\mathcal{X}}^s$ given in (8) as the feasible set in a projected gradient descent algorithm may appear intractable because the constraint $\tilde{a}_i^\top x \leq b$ has to be satisfied for all $\tilde{a}_i \in \mathcal{C}_i(\delta)$. However, using the structure of $\mathcal{C}_i(\delta)$, it can be shown that (Lattimore and Szepesvári 2020, Chapter 19) $\hat{\mathcal{X}}^s$ has a more tractable representation as follows

$$\hat{\mathcal{X}}^s = \{x \in \mathbb{R}^d : \hat{a}_i^\top x + \beta_{T_0}(\delta)\|x\|_{V_{T_0}^{-1}} \leq b_i, \forall i \in [1, m]\}. \tag{9}$$

We will use the above representation, both for implementing our algorithm and analyzing its regret guarantees.

3.3 Online Projected Gradient Descent

After the initial safe exploration for the first T_0 time steps and computing the conservative safe set $\hat{\mathcal{X}}^s$, the SO-PGD algorithm performs online projected gradient descent for $t \in [T_0 + 1, T]$ by treating $\hat{\mathcal{X}}^s$ as the feasible set. Formally, the SO-PGD algorithm takes the sequence of actions $\{x_t, t \in [T_0 + 1, T]\}$ given by

$$x_{t+1} = \Pi_{\hat{\mathcal{X}}^s}(x_t - \eta\nabla f_t(x_t)). \tag{10}$$

Since $\hat{\mathcal{X}}^s$ is a subset of the true safe set $\mathcal{X}^s$, the sequence of actions taken by the SO-PGD algorithm is safe by definition.

3.4 Main Result

We now give the main result of our paper.

Theorem 1. *Let Assumptions 1-4 hold. Consider the SO-PGD algorithm with γ as specified in Lemma 1, $\eta = 2L/G\sqrt{T}$ and $T_0 = T^{2/3}$. Let $\{x_t, t \in [1,T]\}$ be the sequence of actions generated by the SO-PGD algorithm. Then, for any $T \geq \left(\frac{\sqrt{8}\beta_T(\delta)L}{\gamma\sigma\Delta^s}\right)^3$, with a probability greater than $(1-\delta)$, we have*

$$x_t \in \mathcal{X}^s, \ \forall t \in [1,T], \ \text{and}$$

$$R(T) \leq 2LGT^{2/3} + 2LG\sqrt{T} + \frac{LG\sqrt{8d}\beta_T(\delta)}{C(A,b)\sqrt{\gamma^2\sigma_\zeta^2}}T^{2/3}, \tag{11}$$

where $C(A,b)$ is a positive constant that depends only on the matrix A and vector b.

Remark 1. Theorem 1 guarantees that the SO-PGD algorithm achieves $O(T^{2/3})$ regret , excluding the $O(\log T)$ factor resulting from $\beta_T(\delta)$. This is similar to the $O(T^{2/3})$ regret guarantee obtained in (Amani, Alizadeh, and Thrampoulidis 2019) for the safe linear bandits problem. We emphasize that the $O(T^{1/2})$ regret guarantees for safe linear bandits obtained in (Amani, Alizadeh, and Thrampoulidis 2019; Khezeli and Bitar 2020) require additional assumption. In particular, they use the knowledge of a lower bound on the distance between the optimal action and the boundary of the safe set. This is not a meaningful assumption in the OCO setting with arbitrarily-varying cost functions. Designing an algorithm that can achieve a better regret without any additional assumptions is an exciting open question.

4 Regret Analysis

We analyze the regret of the SO-PGD algorithm by decomposing it into three terms as follows:

$$R(T) = \underbrace{\sum_{t=1}^{T_0} f_t(x_t) - f_t(x^\star)}_{\text{Term I}} + \underbrace{\sum_{t=T_0+1}^{T} f_t(x_t) - f_t(\hat{x}^\star)}_{\text{Term II}} + \underbrace{\sum_{t=T_0+1}^{T} f_t(\hat{x}^\star) - f_t(x^\star)}_{\text{Term III}}, \tag{12}$$

where $x^\star = \arg\min_{x\in\mathcal{X}^s} \sum_{t=1}^{T} f_t(x)$ is the optimal action in hindsight with respect to the true safe set $\mathcal{X}^s$ and $\hat{x}^\star = \Pi_{\hat{\mathcal{X}}^s}(x^\star)$ is the projection of $x^\star$ to the conservative safe set $\hat{\mathcal{X}}^s$. The first term accounts for the regret due to the safe exploration phase. The second term characterizes the regret of a standard online projected gradient descent algorithm with respect to the conservative safe set $\hat{\mathcal{X}}^s$. The third term accounts for the error due to using the conservative safe set $\hat{\mathcal{X}}^s$ in the online projected gradient descent instead of the true safe set $\mathcal{X}^s$. We separately analyze the regret of each term and show that the regret is $O(T^{2/3})$.

4.1 Regret of Term I

We bound Term I as follows:

$$\sum_{t=1}^{T_0} f_t(x_t) - f_t(x^\star) \leq \sum_{t=1}^{T_0} G\,\|x_t - x^\star\| \leq 2LGT_0, \tag{13}$$

where the first inequality is from Assumption 1 and the second inequality is by Assumption 2. Now, by selecting T_0 as $T^{2/3}$ as specified in Theorem 1, the regret due to Term I will be $O(T^{2/3})$.

4.2 Regret of Term II

We bound this term using the online projected gradient descent analysis (Hazan 2016) with respect to the estimated safe $\hat{\mathcal{X}}^s$. The regret due to Term II is given by the following proposition.

Proposition 1. *Let Assumptions 1 and 2 hold. Let the learning rate be $\eta = 2L/G\sqrt{T}$. Then,*

$$\sum_{t=T_0+1}^{T} f_t(x_t) - f_t(\hat{x}^\star) \ \leq 2LGT^{1/2}. \tag{14}$$

So, the regret due to Term II will be $O(T^{1/2})$, which is order-wise smaller than the regret due to Term I.

4.3 Regret of Term III

The key step here is to bound $\|\hat{x}^\star - x^\star\|$ as a (decreasing) function of T_0. We can then use the fact that $T_0 = T^{2/3}$ to get the net regret due to this term.

We start by making use of the 'shrunk polytope' idea used in (Fereydounian et al. 2020). Consider the 'shrunk polytope' $\mathcal{X}^s_{\text{in}}$ defined as

$$\mathcal{X}^s_{\text{in}} = \{x \in \mathbb{R}^d : a_i^\top x + \tau_{\text{in}} \leq b_i, \forall i \in [1,m]\}, \tag{15}$$

where τ_{in} is a positive scalar. It is straight forward to note that if τ_{in} is smaller than some constant, $\mathcal{X}^s_{\text{in}}$ will be non-empty and will be a 'shrunk version' of $\mathcal{X}^s$. More precisely, $\mathcal{X}^s_{\text{in}}$ will be a closed polytope with its faces parallel to the faces of $\mathcal{X}^s$, and will be a strict subset of $\mathcal{X}^s$. The key objective for defining this 'shrunk polytope' is to characterize the distance $\|\Pi_{\mathcal{X}^s_{\text{in}}}(x^\star) - x^\star\|$ in terms of τ_{in}, which will then be used to bound the distance $\|\Pi_{\hat{\mathcal{X}}^s}(x^\star) - x^\star\| = \|\hat{x}^\star - x^\star\|$. Note that, our algorithm, however, will not be able to (and does not need to) compute $\mathcal{X}^s_{\text{in}}$ because a_is are unknown. We are using $\mathcal{X}^s_{\text{in}}$ only for the purpose of regret analysis.

We will use the following result from (Fereydounian et al. 2020) to characterize the distance $\|\Pi_{\mathcal{X}^s_{\text{in}}}(x^\star) - x^\star\|$.

Lemma 4 (Lemma 1 in (Fereydounian et al. 2020))**.** *Consider a positive constant τ_{in} such that $\mathcal{X}^s_{\text{in}}$ is non-empty. Then, for any $x \in \mathcal{X}^s$,*

$$\|\Pi_{\mathcal{X}^s_{\text{in}}}(x) - x\| \leq \frac{\sqrt{d}\tau_{\text{in}}}{C(A,b)}, \tag{16}$$

where $C(A,b)$ is a positive constant that depends only on the matrix A and the vector b.

We will now show that the shrunk polytope $\mathcal{X}_{\text{in}}^s$ is non-empty and is a subset of the conservative safe set $\hat{\mathcal{X}}^s$ for $\tau_{\text{in}} = 2\beta_{T_0}(\delta)L/\sqrt{\lambda_{\min}(V_{T_0})}$. This also will immediately imply that $\left\|\Pi_{\hat{\mathcal{X}}^s}(x^\star) - x^\star\right\| \leq \left\|\Pi_{\mathcal{X}_{\text{in}}^s}(x^\star) - x^\star\right\|$. We state this result formally below.

Lemma 5. *Let Assumptions 2 and 3 hold. Let* $\tau_{\text{in}} = 2\beta_{T_0}(\delta)L/\sqrt{\lambda_{\min}(V_{T_0})}$ *and* $T_0 \geq \frac{8\beta_T^2(\delta)L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}$. *Then,* $\mathcal{X}_{\text{in}}^s$ *is non-empty and* $\mathcal{X}_{\text{in}}^s \subseteq \hat{\mathcal{X}}^s$, *with a probability greater than* $(1-2\delta)$. *Moreover,* $\left\|\Pi_{\hat{\mathcal{X}}^s}(x^\star) - x^\star\right\| \leq \left\|\Pi_{\mathcal{X}_{\text{in}}^s}(x^\star) - x^\star\right\|$ *with a probability greater than* $(1-2\delta)$.

Using the above lemma, we can now characterize the regret due to Term III as stated in the proposition below.

Proposition 2. *Let Assumptions 1 - 3 hold. Then, for* $T_0 \geq \frac{8\beta_T^2(\delta)L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}$, *with a probability greater than* $(1-2\delta)$,

$$\sum_{t=T_0+1}^{T} f_t(\hat{x}^\star) - f_t(x^\star) \leq \frac{LG\sqrt{8d}\beta_T(\delta)}{C(A,b)\sqrt{\gamma^2\sigma_\zeta^2}} \frac{T}{\sqrt{T_0}}. \quad (17)$$

Note that, when we use $T_0 = T^{2/3}$ in the above result, we get the regret due to Term III as $O(T^{2/3})$.

The proof of our main theorem can now be obtained by adding the regret due to Terms I, II, and III.

5 Simulation Results

In this section, we analyze the performance of our SO-PGD algorithm through experiments in two different settings.

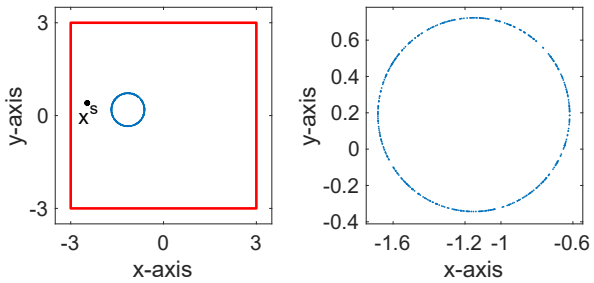
Experiment Setting: We consider a closed polytope of the form $\mathcal{X}^s = \{x \in \mathbb{R}^2 : -x_{\max} \leq x_i \leq x_{\max}, i = 1, 2\}$ as the safe set. It is straight forward to see that the corresponding parameters are $A = [1, 0; -1, 0; 0, 1; 0, -1]$ and $b = x_{\max} \times [1; 1; 1; 1]$. We consider two sequences of functions, $f_{1,t}$ and $f_{2,t}$, given by

$$f_{1,t}(x) = c_t \cdot (\sum_{i=1}^{d} x_i) + 1, \quad f_{2,t}(x) = \frac{1}{2}\|x - c_t\bar{x}\|_2^2,$$

where c_t is a real number drawn i.i.d. from the set $[c_{\text{lower}}, c_{\text{upper}}]$. We select c_{lower} and c_{upper} appropriately from $\{0.5, 1, 1.5, 2\}$ for different experiment settings. For $f_{2,t}$, $\bar{x}$ is randomly sampled from a standard Gaussian distribution, then normalized and scaled by 2.5. The constraint noise sequence w_ts are i.i.d. Gaussian with zero mean and covariance matrix $10^{-3}I$. Note that $f_{1,t}$s are linear function $f_{2,t}$s are a 1-strongly convex function.

For a fixed T, we first generate the sequence $c_t, t \in [1, T]$. Then, we find the optimal action in hindsight, $x_i^\star = \arg\min_{x \in \mathcal{X}^s} \sum_{t=1}^{T} f_{i,t}(x)$, $i = 1, 2$, using a standard non-linear optimization function like `fmincon` from `MATLAB`.

We choose $\lambda = 0.5$ and $\delta = 10^{-3}$. Exploration noise is ζ_ts are generated according to a standard Gaussian distribution and then normalized. The safe baseline action is selected randomly from the set $\mathcal{X}^s$. We run the experiments with $T = 10^6$ and $T_0 = T^{2/3} = 10^4$. We emphasize that for these values, the condition $T_0 \geq \frac{8\beta^2 L^2}{\gamma^2\sigma_\zeta^2(\Delta^2)^2}$ specified in Theorem 1 is satisfied.



(a) (b)

Figure 1: (a) Shows the safe set $\mathcal{X}^s$ (marked in red), safe baseline action x^s, and actions taken during the safe exploration phase (marked in blue). All the actions lie inside $\mathcal{X}^s$. (b) Shows zoom-in view of the actions shown in (a).

Safe exploration: Figure 1 shows the safe baseline action and the actions taken during the safe exploration phase. As guaranteed by Lemma 1, all actions are strictly inside the safe set.

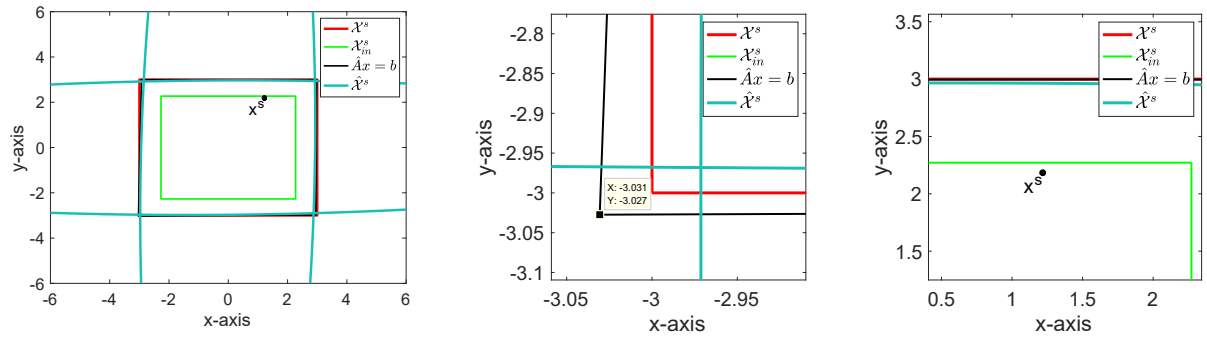
Conservative safe set estimation: Fig. 2(a) shows the true safe set ($\mathcal{X}^s$), the conservative safe set estimate ($\hat{\mathcal{X}}^s$), and the 'shrunk polytope' ($\mathcal{X}_{\text{in}}^s$). We also show the polytope obtained using the naive least squares estimate, $\{x : \hat{A}x \leq b\}$, where $\hat{A}$ is obtained according to (5). Please see that $\mathcal{X}_{\text{in}}^s \subset \hat{\mathcal{X}}^s \subset \mathcal{X}^s$, as guaranteed by our results in Lemma 3 and Lemma 5. It can also be seen that the polytope obtained using the naive least squares estimate need not be a subset of the safe set $\mathcal{X}^s$. We highlight this aspect in Fig. 2(b). So, an OCO algorithm that uses this naive estimate cannot guarantee safety constraint satisfaction at all time steps. Fig. 2(c) also shows that the safe baseline action x^s is inside the 'shrunk polytope' $\mathcal{X}_{\text{in}}^s$, as guaranteed by our theory (see the proof of Lemma 5).

Online gradient descent: Fig. 4 shows the sequence of actions generated by the SO-PGD algorithm in one experiment. We do not plot all the actions, but only a regularly sampled version of the sequence of actions to avoid crowding the plot. Notice that these actions lie inside the safe set $\mathcal{X}^s$

Regret performance: The regret performance of the SO-PGD algorithm is shown in Fig. 3. Instead of plotting regret directly, we plot $R(t)/t$ and $R(t)/t^{2/3}$, where $R(t)$ is the cumulative regret incurred until time t. From the figures, it is easy to observe that $R(t)/t$ goes to zero, ensuring that the regret is indeed sublinear. Also, $R(t)/t^{2/3}$ converges to a constant value, indicating that the regret of the SO-PGD algoirthm is indeed $O(T^{2/3})$, as guaranteed by Theorem 1.

6 Conclusion

In this work, we addressed the problem of safe online convex optimization, where the action at each time step must satisfy a set of linear safety constraints. The parameters that



(a) (b) (c)

Figure 2: (a) Shows the true safe set ($\mathcal{X}^s$), the conservative safe set estimate ($\hat{\mathcal{X}}^s$), the 'shrunk polytope' ($\mathcal{X}^s_{\text{in}}$), and the naive least squares estimate of the safe set ($\hat{A}x = b$). (b) Shows a zoomed-in view of the third quadrant of (a). (c) Shows a zoomed-in view of the region around x^s to show that it lies inside $\mathcal{X}^s_{\text{in}}$.

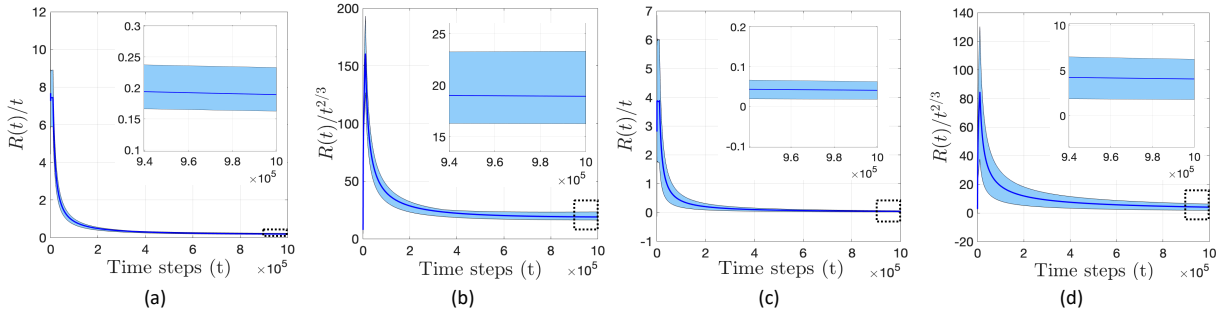


Figure 3: (a) $R(t)/t$ vs t for f_1. (b) $R(t)/t^{2/3}$ vs t for f_1. (c) $R(t)/t$ vs t for f_2. (d) $R(t)/t^{2/3}$ vs t for f_2. All the plots are averaged over 6 random realizations. The light blue region shows the error (max - min) in mean value. The subplots in the top right corner are zoomed-in view into the final 60000 time steps. $R(t)/t$ decays to zero for both f_1 and f_2. $R(t)/t^{2/3}$ converges to a constant value ~ 20 for f_1, and ~ 4 for f_2.

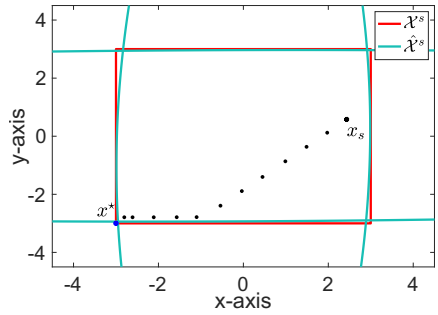


Figure 4: The (sampled) sequence of actions taken by the SO-PGD algorithm in the online projected gradient descent phase. All the actions lie inside the true safe set $\mathcal{X}^s$.

specify the linear safety constraints are unknown to the algorithm. We proposed an algorithm called SO-PGD algorithm to solve this problem. Our algorithm comprises of two phases, a safe exploration phase to estimate the unknown safe set and an online gradient descent phase for online optimization. We showed that by carefully balancing the duration of the exploration phase and online optimization phase, the SO-PGD algorithm can achieve $O(T^{2/3})$ regret while satisfying the safety constraints at all times step, with high probability. To the best of our knowledge, this is the first such result in the OCO literature, even in a setting with liner constraints.

In the future, we plan to extend our results to develop projection-free safe OCO algorithms. We will also investigate if it is possible to achieve $O(T^{1/2})$ regret with no constraint violation, without making additional strong assumption.

Acknowledgments

Dileep Kalathil gratefully acknowledges funding from the U.S. National Science Foundation (NSF) grants NSF-EAGER-1839616, NSF-CRII-CPS-1850206 and NSF-CAREER-EPCN-2045783.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24: 2312–2320.

Amani, S.; Alizadeh, M.; and Thrampoulidis, C. 2019. Linear stochastic bandits under safety constraints. In *Advances in Neural Information Processing Systems*, 9256–9266.

Åström, K. J.; and Murray, R. M. 2010. *Feedback systems*. Princeton university press.

Cao, X.; and Liu, K. R. 2018. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Transactions on automatic control*, 64(7): 2665–2680.

Cesa-Bianchi, N.; Long, P. M.; and Warmuth, M. K. 1996. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Transactions on Neural Networks*, 7(3): 604–619.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Chaudhary, S.; and Kalathil, D. 2021. Safe Online Convex Optimization with Unknown Linear Safety Constraints. *arXiv preprint arXiv:2111.07430*.

Chen, T.; Ling, Q.; and Giannakis, G. B. 2017. An online convex optimization approach to proactive network resource allocation. *IEEE Transactions on Signal Processing*, 65(24): 6350–6364.

Dobbe, R.; Hidalgo-Gonzalez, P.; Karagiannopoulos, S.; Henriquez-Auba, R.; Hug, G.; Callaway, D. S.; and Tomlin, C. J. 2020. Learning to control in power systems: Design and analysis guidelines for concrete safety problems. *Electric Power Systems Research*, 189: 106615.

Fereydounian, M.; Shen, Z.; Mokhtari, A.; Karbasi, A.; and Hassani, H. 2020. Safe Learning under Uncertain Objectives and Constraints. *arXiv preprint arXiv:2006.13326*.

Gordon, G. J. 1999. Regret bounds for prediction problems. In *Proceedings of the twelfth annual conference on Computational learning theory*, 29–40.

Hazan, E. 2016. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, 2(3-4): 157–325.

Ibaraki, T.; and Katoh, N. 1988. *Resource allocation problems: algorithmic approaches*. MIT press.

Khezeli, K.; and Bitar, E. 2020. Safe linear stochastic bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10202–10209.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Luong, N. C.; Hoang, D. T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.-C.; and Kim, D. I. 2019. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4): 3133–3174.

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.

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

Shalev-Shwartz, S. 2011. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2): 107–194.

Sun, W.; Dey, D.; and Kapoor, A. 2017. Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*, 3280–3288. PMLR.

Tropp, J. A. 2015. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2): 1–230.

Usmanova, I.; Krause, A.; and Kamgarpour, M. 2019. Safe Convex Learning under Uncertain Constraints. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2106–2114.

Wei, X.; Yu, H.; and Neely, M. J. 2020. Online primal-dual mirror descent under stochastic constraints. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2): 1–36.

Yi, X.; Li, X.; Xie, L.; and Johansson, K. H. 2020. Distributed online convex optimization with time-varying coupled inequality constraints. *IEEE Transactions on Signal Processing*, 68: 731–746.

Yu, H.; Neely, M.; and Wei, X. 2017. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, 1428–1438.

Yu, H.; and Neely, M. J. 2020. A Low Complexity Algorithm with $O(\sqrt{T})$ Regret and $O(1)$ Constraint Violations for Online Convex Optimization with Long Term Constraints. *Journal of Machine Learning Research*, 21(1): 1–24.

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

A Appendix

A.1 Preliminaries

We use following well known result from linear bandits literature.

Theorem 2 (Theorem 2,(Abbasi-Yadkori, Pál, and Szepesvári 2011)). *Let $\{F_t\}_{t=0}^{\infty}$ be a filtration. Let $\{\eta_t\}_{t=1}^{\infty}$ be a real valued stochastic process such that η_t is F_t-measurable and η_t is conditionally R-sub-Guassian for some $R \geq 0$. Let $\{X_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process such that X_t is F_{t-1}-measurable. Let V_t be defined as $\lambda I + \sum_{t=1}^{t} X_t X_t^\top$ for $\lambda > 0$. Let, $Y_t = a^\top x + \eta_t$. Let $\hat{a}_t = V_t^{-1} \sum_{i=1}^{t} Y_i X_i$ be the ℓ_2-regularized least squares estimate of a. Assume that $\|a\| \leq L_A$ and $\|X_t\| \leq L, \forall t$. Then, for any $\delta > 0$, with a probability at least $1-\delta$, the true parameter a lies in the set*

$$C_t = \left\{ a \in \mathbb{R}^d : \|\hat{a} - a\|_{V_t} \leq R\sqrt{d \log\left(\frac{1 + tL^2/\lambda}{\delta}\right)} + \sqrt{\lambda} L_A \right\}, \tag{18}$$

for all $t \geq 1$.

We will use the following results on matrix Chernoff inequality (Tropp 2015, Theorem 5.1.1.).

Theorem 3 (Theorem 5.1.1. (Tropp 2015)). *Consider a finite sequence $\{X_k\}$ of independent, random, symmetric matrices with a common dimension d. Assume that $\lambda_{\min}(X_k) \geq 0$ and $\lambda_{\max}(X_k) \leq L, \forall k$. Introduce the random matrix $Y = \sum_k X_k$. Define the minimum eigenvalue $\mu_{\min}$ of the expectation $\mathbb{E}[Y]$ as $\mu_{\min} = \lambda_{\min}(\mathbb{E}[Y]) = \lambda_{\min}(\sum_k \mathbb{E}[X_k])$. then,*

$$\mathbb{P}(\lambda_{\min}(Y) \leq \epsilon \mu_{\min}) \leq d\, \mathrm{e}^{-(1-\epsilon)^2 \mu_{\min}/2L} \text{ for any } \epsilon \in (0,1). \tag{19}$$

A.2 Proof of Lemma 1

Proof. For any $t \in [1, T_0]$, and for any $i \in [1, m]$, we have

$$\begin{aligned} a_i^\top x_t &\overset{(i)}{=} a_i^\top((1-\gamma)x^s + \gamma\zeta_t) \\ &= (1-\gamma)a_i^\top x^s + \gamma a_i^\top \zeta_t = (1-\gamma)b_i^s + \gamma a_i^\top \zeta_t \overset{(ii)}{\leq} (1-\gamma)b_i^s + \gamma L_A \min\{1, L\} \end{aligned}$$

Here, we get (i) by the definition of exploration action (4), and (ii) by using the fact that $\max_i \|a_i\| \leq L_A$ and $\|\zeta_t\| \leq \min\{1, L\}$. Now, for x_t to satisfy the safety constraint, it is sufficient to have $(1-\gamma)b_i^s + \gamma L_A \min\{1, L\} \leq b_i$, or equivalently, $\gamma(L_A \min\{1, L\} - b_i^s) \leq (b_i - b_i^s)$. This leads to the sufficient condition $\gamma L_A \leq \min_i\ (b_i - b_i^s) = \Delta^s$. □

A.3 Proof of Lemma 2 and Lemma 3

Proof of Lemma 2. Using Theorem 2, for any $i \in [1, m]$, we get

$$\mathbb{P}(a_i \in \mathcal{C}_i(\delta)) \geq 1 - \delta/m.$$

Now, we get the desired result by applying union bound. □

Proof of Lemma 3. From Lemma 2, $a_i \in \mathcal{C}_i(\delta)$ for all $i \in [1, m]$ with a probability greater than $(1-\delta)$. Then, by definition, for any $x \in \hat{\mathcal{X}}^s$, we have $a_i^\top x \leq b_i$ for all $i \in [1, m]$, with probability greater than $1-\delta$. So, $\hat{\mathcal{X}}^s \subseteq \mathcal{X}^s$ with probability at least $1-\delta$. □

A.4 Proof of Proposition 1

This results follows from the standard regret analysis of online projected gradient descent algorithm (Hazan 2016, Theorem 3.1). We reproduce the result here for completeness.

Proof. By convexity of the function f_t

$$f_t(x_t) - f_t(\hat{x}^\star) \leq \nabla f_t(x_t)^\top (x_t - \hat{x}^\star). \tag{20}$$

We will now upper bound $\nabla f_t(x_t)^\top (x_t - \hat{x}^\star)$ as follows:

$$\begin{aligned} \|x_{t+1} - \hat{x}^\star\|^2 &= \left\| \prod_{\hat{\mathcal{X}}^s} (x_t - \eta \nabla f_t(x_t)) - \hat{x}^\star \right\|^2 \leq \|x_t - \eta \nabla f_t(x_t) - \hat{x}^\star\|^2 \\ &= \|x_t - \hat{x}^\star\|^2 + \eta^2 \|\nabla f_t(x_t)\|^2 - 2\eta \nabla f_t(x_t)^\top (x_t - \hat{x}^\star), \end{aligned}$$

where the first inequality is by the Pythagorean theorem. Rearranging and using Assumption 1, we get

$$2\nabla f_t(x_t)^\top (x_t - \hat{x}^\star)\ \leq \frac{\|x_t - \hat{x}^\star\|^2 - \|x_{t+1} - \hat{x}^\star\|^2}{\eta} + \eta G^2. \tag{21}$$

Using (21) in (20) and taking summation, and using the fact that $\eta = 2L/G\sqrt{T}$ we get

$$\sum_{t=T_0+1}^{T}(f_t(x_t) - f_t(\hat{x}^\star)) \leq \sum_{t=T_0+1}^{T}\nabla f_t(x_t)^\top(x_t - \hat{x}^\star) \leq \sum_{t=T_0+1}^{T}\frac{\|x_t - \hat{x}^\star\|^2 - \|x_{t+1} - \hat{x}^\star\|^2}{2\eta} + \frac{G^2}{2}\sum_{t=T_0+1}^{T}\eta$$
$$\leq \|x_1 - \hat{x}^\star\|^2\frac{1}{2\eta} + \frac{G^2}{2}T\eta \leq 2L^2\frac{1}{\eta} + \frac{G^2}{2}T\eta \leq LG\sqrt{T} + LG\sqrt{T} = 2LG\sqrt{T}. \tag{22}$$

□

A.5 Proof of Lemma 5 and Proposition 2

One key step in proving Lemma 5 and Proposition 2 is to get a high probability lower bound on $\lambda_{\min}(V_{T_0}$. We will use the matrix matrix Chernoff inequality for achieving this. We state this result as a lemma below.

Lemma 6. *For $T_0 \geq \frac{8L^2}{\gamma^2\sigma_\zeta^2}\log\frac{d}{\delta}$, we have*

$$\mathbb{P}(\lambda_{\min}(V_{T_0}) \geq \lambda + 0.5\gamma^2\sigma_\zeta^2 T_0) \geq (1-\delta). \tag{23}$$

Proof. For $t \in [1, T_0]$, $x_t = (1-\gamma)x^s + \gamma\zeta_t$, where ζ_ts are zero mean i.i.d. random vectors such that $\|\zeta_t\| \leq \min\{1, L\}$ and $\mathbb{E}[\zeta_t\zeta_t^\top] = \sigma_\zeta^2 I$. Let $X_t = x_t x_t^\top$. Then, X_t is symmetric and positive semidefinite. So, $\lambda_{\min}(X_t) \geq 0$. Also, $\lambda_{\max}(X_t) \leq \|x_t\|^2 \leq L^2$. We will also get that $\mathbb{E}[X_t] = (1-\gamma)^2 x^s(x^s)^\top + \gamma^2\sigma_\zeta^2 I$.

Let $Y = \sum_{t=1}^{T_0} X_t$ and $\mu_{\min} = \lambda_{\min}(\mathbb{E}[Y])$. Then,

$$\mu_{\min} = \lambda_{\min}(\mathbb{E}[Y]) = \lambda_{\min}(\sum_{t=1}^{T_0}\mathbb{E}[X_t]) = \lambda_{\min}(T_0((1-\gamma)^2 x^s(x^s)^\top + \gamma^2\sigma_\zeta^2 I)) \geq \gamma^2\sigma_\zeta^2 T_0. \tag{24}$$

Now, using the matrix Chernoff inequality stated in Theorem 3, and the above inequality (24), we get

$$\mathbb{P}(\lambda_{\min}(Y) \leq \epsilon\gamma^2\sigma_\zeta^2 T_0) \leq \mathbb{P}(\lambda_{\min}(Y) \leq \epsilon\mu_{\min}) \leq d\exp\left(-\frac{(1-\epsilon)^2\mu_{\min}}{2L^2}\right) \leq d\exp\left(-\frac{(1-\epsilon)^2\gamma^2\sigma_\zeta^2 T_0}{2L^2}\right) \tag{25}$$

For $\epsilon = 1/2$, with $T_0 \geq \frac{8L^2}{\gamma^2\sigma_\zeta^2}\log\frac{d}{\delta}$, we get

$$\mathbb{P}(\lambda_{\min}(Y) \geq 0.5\gamma^2\sigma_\zeta^2 T_0) \geq (1-\delta). \tag{26}$$

Since $V_{T_0} = \lambda I + \sum_{t=1}^{T_0} x_t x_t^\top = \lambda I + Y$, we have $\lambda_{\min}(V_{T_0}) \geq \lambda + \lambda_{\min}(Y)$. This will give,

$$\mathbb{P}(\lambda_{\min}(V_{T_0}) \geq \lambda + 0.5\gamma^2\sigma_\zeta^2 T_0) \geq (1-\delta). \tag{27}$$

□

Consider the events

$$\mathcal{E}_A = \{a_i \in \mathcal{C}_i(\delta), \forall i \in [1,m]\},\ \ \mathcal{E}_\lambda = \{\lambda_{\min}(V_{T_0}) \geq \lambda + 0.5\gamma^2\sigma_\zeta^2 T_0\},\ \ \mathcal{E} = \mathcal{E}_A \cap \mathcal{E}_\lambda. \tag{28}$$

From Lemma 2, $\mathbb{P}(\mathcal{E}_A) \geq (1-\delta)$. From Lemma 6, with $T_0 \geq \frac{8L^2}{\gamma^2\sigma_\zeta^2}\log\frac{d}{\delta}$, $\mathbb{P}(\mathcal{E}_\lambda) \geq (1-\delta)$. Then, using union bound, $\mathbb{P}(\mathcal{E}) \geq 1 - 2\delta$. Our analysis for the proof of Lemma 5 and Proposition 2 will be conditioned on the event $\mathcal{E}$. So, they will be true with a probability greater than $(1-2\delta)$.

We now give the proof of Lemma 5.

Proof of Lemma 5. To show that $\mathcal{X}_{\text{in}}^s$ is non-empty, we will show that x^s is an element of $\mathcal{X}_{\text{in}}^s$ for $T_0 \geq \frac{8\beta_T^2(\delta)L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}$. For x^s to be an element of $\mathcal{X}_{\text{in}}^s$, we need

$$a_i^\top x_s + \tau_{\text{in}} \leq b_i, \forall i \in [1,m] \quad \Longrightarrow \quad \tau_{\text{in}} \leq \min_i\ (b_i - a_i^\top x_s) = \min_i\ (b_i - b_i^s) = \Delta^s.$$

For $\tau_{\text{in}} = 2\beta_{T_0}(\delta)L/\sqrt{\lambda_{\min}(V_{T_0})}$, this is equivalent to satisfying the condition $\lambda_{\min}(V_{T_0}) \geq \frac{4\beta_{T_0}^2(\delta)L^2}{(\Delta^s)^2}$. Now, conditioned on the event $\mathcal{E}$, this inequality is satisfied with a probability greater than $(1-2\delta)$ if $\lambda + 0.5\gamma^2\sigma_\zeta^2 T_0 \geq \frac{4\beta_{T_0}^2(\delta)L^2}{(\Delta^s)^2}$, which is guaranteed for any T_0 such that

$$T_0 \geq \frac{8\beta_T^2(\delta)L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}. \tag{29}$$

Please note that the above lower bound on T_0 also satisfies the lower bound condition for the result of Lemma 6 to be true when Δ^s is small or T is large, which is typically the case. So, when T_0 satisfies the condition (29), $x^s \in \mathcal{X}^s_{\text{in}}$, and hence $\mathcal{X}^s_{\text{in}}$ is non-empty.

To show that $\mathcal{X}^s_{\text{in}} \subset \hat{\mathcal{X}}^s$, consider an arbitrary $x \in \mathcal{X}^s_{\text{in}}$. Then, by definition, $a_i^\top x + \frac{2\beta_{T_0}(\delta)L}{\sqrt{\lambda_{\min}(V_{T_0})}} \leq b_i$. Now,

$$\begin{aligned}
\hat{a}_i^\top x + \beta_{T_0}(\delta)\left\|x\right\|_{V_{T_0}^{-1}} &= a_i^\top x + (\hat{a}_i^\top x - a_i^\top x) + \beta_{T_0}(\delta)\left\|x\right\|_{V_{T_0}^{-1}} \leq a_i^\top x + \left\|\hat{a}_i - a_i\right\|_{V_{T_0}} \left\|x\right\|_{V_{T_0}^{-1}} + \beta_{T_0}(\delta)\left\|x\right\|_{V_{T_0}^{-1}} \\
&\overset{(i)}{\leq} a_i^\top x + 2\beta_{T_0}(\delta)\left\|x\right\|_{V_{T_0}^{-1}} \\
&\leq a_i^\top x + \frac{2\beta_{T_0}(\delta)\left\|x\right\|_2}{\sqrt{\lambda_{\min}(V_{T_0})}} \overset{(ii)}{\leq} a_i^\top x + \frac{2\beta_{T_0}(\delta)L}{\sqrt{\lambda_{\min}(V_{T_0})}} \leq b_i,
\end{aligned} \tag{30}$$

where we get (i) conditioned on the event $\mathcal{E}$ and (ii) by using the fact that $\left\|x\right\|_2 \leq L$. This implies that $x \in \hat{\mathcal{X}}^s$. Since $x \in \mathcal{X}^s_{\text{in}}$ is arbitrary, we get $\mathcal{X}^s_{\text{in}} \subset \hat{\mathcal{X}}^s$. This also immediately implies that $\left\|\Pi_{\hat{\mathcal{X}}^s}(x^\star) - x^\star\right\| \leq \left\|\Pi_{\mathcal{X}^s_{\text{in}}}(x^\star) - x^\star\right\|$. □

We now give the proof of Proposition 2

Proof of Proposition 2. Conditioned on the event $\mathcal{E}$,

$$\begin{aligned}
\sum_{t=T_0+1}^{T} f_t(\hat{x}^\star) - f_t(x^\star) &\overset{(i)}{\leq} G(T-T_0)\left\|\hat{x}^\star - x^\star\right\| = G(T-T_0)\left\|\Pi_{\hat{\mathcal{X}}^s}(x^\star) - x^\star\right\| \\
&\overset{(ii)}{\leq} GT\left\|\Pi_{\mathcal{X}^s_{\text{in}}}(x^\star) - x^\star\right\| \overset{(iii)}{\leq} GT\frac{\sqrt{d}}{C(A,b)}\tau_{\text{in}} \overset{(iv)}{=} GT\frac{\sqrt{d}}{C(A,b)}\frac{2\beta_{T_0}(\delta)L}{\sqrt{\lambda_{\min}(V_{T_0})}},
\end{aligned}$$

where (i) is by using Assumption 1, (ii) from Lemma 5, (iii) is by using Lemma 4, and (iv) is by applying the value of τ_{in} used in Lemma 5.

Also, conditioned on the event $\mathcal{E}$, we have $\lambda_{\min}(V_{T_0}) \geq \lambda + 0.5\gamma^2\sigma_\zeta^2 T_0$. Using this in the above inequality, we get

$$\sum_{t=T_0+1}^{T} f_t(\hat{x}^\star) - f_t(x^\star) \leq GT\frac{\sqrt{d}}{C(A,b)}\frac{2\beta_{T_0}(\delta)L}{\sqrt{\lambda + 0.5\gamma^2\sigma_\zeta^2 T_0}} \leq GT\frac{\sqrt{d}}{C(A,b)}\frac{2\beta_{T_0}(\delta)L}{\sqrt{0.5\gamma^2\sigma_\zeta^2 T_0}}$$

Reordering the terms, we get the stated result.

□

A.6 Proof of Theorem 1

Proof of Theorem 1. We first prove the safety guarantee. For $t \in [1, T_0]$, $x_t \in \mathcal{X}^s$ by Lemma 1. For $t > T_0$, the SO-PGD algorithm performs online projected gradient descent with respect to the set $\hat{\mathcal{X}}^s$. So $x_t \in \hat{\mathcal{X}}^s$ for $t \in [T_0+1, T]$. Now, by Lemma 3, $\hat{\mathcal{X}}^s \subset \mathcal{X}^s$ with a probability greater than $(1-\delta)$. So, $x_t \in \mathcal{X}^s, \forall t \in [1, T]$, with a probability greater than $(1-\delta)$.

We now prove the regret bound. From the regret decomposition in (12), we have $R(T) = \text{Term I} + \text{Term II} + \text{Term III}$. Using the upper bound for Term I from (13), the upper bound for Term II from Proposition 1, and the upper bound for Term III from Proposition 2, we get

$$R(T) \leq 2LGT_0 + 2LG\sqrt{T} + \frac{LG\sqrt{8d}}{C(A,b)\sqrt{\gamma^2\sigma_\zeta^2}}\frac{\beta_T(\delta)T}{\sqrt{T_0}}, \tag{31}$$

with a probability greater than $(1-2\delta)$, for $T_0 \geq \frac{8\beta_T^2(\delta)L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}$. We will now select $T_0 = T^{2/3}$. To ensure the lower bound condition on T_0 given in Proposition 2, it is sufficient to have $T_0 = T^{2/3} \geq \frac{8\beta_T^2 L^2}{\gamma^2\sigma_\zeta^2(\Delta^s)^2}$, which is equivalent to having

$$T \geq \left(\frac{\sqrt{8}\beta_T(\delta)L}{\gamma\sigma\Delta^s}\right)^3. \tag{32}$$

Now, using $T_0 = T^{2/3}$ in (31), we get

$$R(T) \leq 2LGT^{2/3} + 2LG\sqrt{T} + \frac{LG\sqrt{8d}}{C(A,b)\sqrt{\gamma^2\sigma_\zeta^2}}\beta_T(\delta)T^{2/3}. \tag{33}$$

□

A.7 Additional Simulation Results

Now, we consider another cost function that together with a set of linear inequality constraints of the form $Ax \leq b$ is a representative of problems arising in resource allocation, network scheduling, finance portfolio selection, among others listed in (Yu and Neely 2020; Ibaraki and Katoh 1988). We choose the cost functions $f_{3,t} = c_t^\top x$ as used in (Yu and Neely 2020). Here, $c_t = c_{1,t} + c_{2,t} + c_{3,t}$ to obtain arbitrarily varying $f_{3,t}$. Each component of the term $c_{1,t}$ is uniformly sampled from the interval $\left[-t^{1/10}, +t^{1/10}\right]$. Each component of the term $c_{2,t}$ is uniformly sampled from $[-1, 0]$ for all $t \in [1, 1500] \cup [2000, 3500] \cup [4000, 5000]$, and is otherwise uniformly sampled from $[0, 1]$. To obtain the term $c_{3,t}$, we first obtain a vector p that contains random permutations of numbers in the interval $[1, T]$, where T is the total time steps we run the experiments for. Then, for each $t \in 1, \ldots, T$, $c_{3,t}(i) = (-1)^{p(t)}\, \forall i \in 1, \ldots, d$. We use $d = 2$, and the same constraint polytope formed by $A = [1, 0; -1, 0; 0, 1; 0, -1]$ and $b = x_{\max} \times [1; 1; 1; 1]$, where we choose $x_{\max} = 3$. We now include regret plots for $T = 10^5$ time steps for f_3 in Fig. 5.

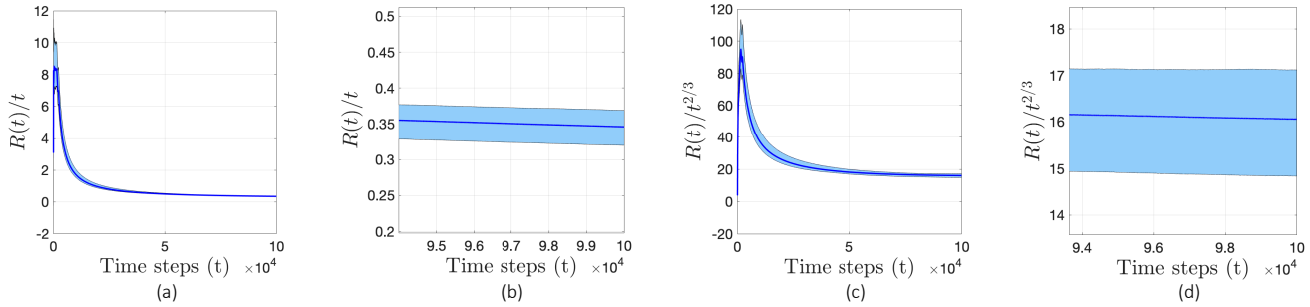


Figure 5: (a) $R(t)/t$ vs t for f_3. (b) Zoomed in version of $R(t)/t$ vs t for f_3. (c) $R(t)/t^{2/3}$ for f_3. (d) Zoomed in version of $R(t)/t^{2/3}$ vs t for f_3. All the plots are averaged over 4 random realizations. The light blue region shows the error (max - min) in mean value.

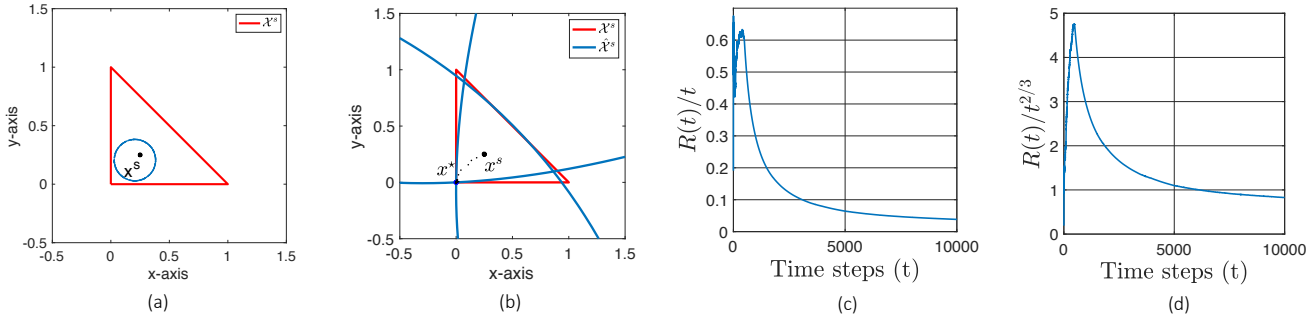


Figure 6: (a) Shows $\mathcal{X}^s$ and exploratory actions. (b) Shows estimated safe set, it is the interior of blue curves. The black dots from x^s to $x^\star$ denote the optimization trajectory. (c) $R(t)/t$ for f_3 for triangular safe set. (d) $R(t)/t^{2/3}$ for f_3 for triangular safe set.

Different safe set We perform an additional experiment to show that our algorithm works well independent of the shape of closed convex safe action set. We choose cost functions of the form f_3 described above and a triangular true safe set $\mathcal{X}^s = \{x \in \mathbb{R}^2 : Ax \leq b\}$ such that $A = [1,\ 1;\ -1,\ 0;\ 0,\ -1]$ and $b = [1,\ 0,\ 0]^\top$. We choose $x^s = [0.25, 0.25]^\top$. We run SO-PGD for this setup for $T = 10^4$ time steps. The results from this experiment are recorded in Fig. 6. We observe, like before, that all the exploratory actions (represented by blue circular region around x^s) are safe (see Fig. 6, (a)). The whole optimization trajectory lies inside $\mathcal{X}^s$ (see Fig. 6, (b)). The decay of regret with respect to t is plotted in Fig. 6, (c), and the decay with respect to $t^{2/3}$ is plotted in Fig. 6, (d). As expected $R(t)/t \to 0$ and $R(t)/t^{2/3}$ tends to a constant value.