Dissertation ETH Zürich No. 28605

Safe Learning via Constrained Stochastic Optimization

A dissertation submitted to attain the degree of

Doctor of Sciences of ETH Zürich

(Dr. Sc. ETH Zürich)

presented by

Ilnura Usmanova

M. Sc., Institut politechnique de Grenoble, France

born 27.04.1994 in Almaty, Kazakhstan
citizen of Russia

accepted on the recommendation of
Prof. Dr. Maryam Kamgarpour, examiner
Prof. Dr. Andreas Krause, co-examiner
Prof. Dr. Amin Karbasi, co-examiner

2022

ETH Zürich
IfA - Automatic Control Laboratory
ETL, Physikstrasse 3
8092 Zurich, Switzerland

To the memory of my dear grandmother Zahira, to my family, and to my husband Mikhail

Acknowledgements

Five years ago, I was lucky to get accepted for a Ph.D. position in Prof. Dr. Maryam Kamgarpour's group at ETH Zurich. Since then, I started my research journey in the friendly and supportive atmosphere of many talented and bright people from the IfA lab and LAS group. I have never regretted my decision since then. The journey was challenging but rewarding and shaped me both professionally and personally. Although a significant part of my Ph.D. studies was during the pandemic, I still had a chance to travel to conferences and doctoral schools, collaborate with excellent professionals, and enjoy exciting events within the IfA and the Sycamore labs and LAS group.

I want to express my highest gratitude to my supervisor, **Prof. Dr. Maryam Kamgarpour**, for her careful support while supervising me and for invaluable science and life advice. I also want to thank her for the supportive and encouraging atmosphere she maintains in her group and for the academic freedom she allows for. I express my immense gratitude to **Prof. Dr. Andreas Krause** for co-supervising my Ph.D. studies, for all the careful support and great ideas he provided during our collaboration, and in general, for the opportunity to work with him and his group. I would also like to thank **Prof. Dr. Amin Karbasi** for kindly agreeing to become a committee member for examining this thesis. I am very grateful to **Prof. Dr. Kfir Levy**, for the fruitful discussions and collaborations and an invitation to a postdoc position in his lab at Technion. Furthermore, I thank **Yarden As** for being a brilliant and motivated student from whom I learned much about reinforcement learning and for being a driving force of our reinforcement learning project, which would not be possible without **Sebastian Curi**. Also, I thank **Anastasiia Makarova** for involving me in the risk-averse bandits project, which would not be possible without **Prof. Dr. Ilija Bogunovic**.

I thank **Dr. Anna Volokitin** for numerous proofreads of my writings. I thank all the members of **IfA** lab for being a driving force of an encouraging and warm atmosphere inside the lab. I am very grateful to my teammates **Pier Giuseppe Sessa**, **Dr. Orcun Karaca**, **Dr. Luca Furieri**, and **Yimeng Lu** for being very supportive and friendly. I thank **Dr. Alisa Rupenyan** for the many exciting discussions, inspiring me about the various research applications, and proofreading parts of this dissertation. I thank my roommates **Jeremy** and **Xavier** for numerous interesting discussions. Thanks to **Dr. Goran Banjac** for allowing me to teach your optimization course, which was extremely exciting, and **Liviu** for incredible support during my teaching duties. Thanks to **Samuel** for organizing informal IfA events. I thank the members of **LAS** for the friendly and open atmosphere I could enjoy during our group meetings. Thanks to **Dr. Johannes Kirschner** for being so open and responsive which was extremely helpful during my work

on experiments. Thanks to **Mohammad Reza Karimi** for inspiring discussions and proofreading parts of this thesis. Thank you **Dr. Tony Wood**, **Anna**, **Andreas**, and **Ting Ting**, for the warm and motivating atmosphere you maintain at Sycamore lab.

A special thanks go to **Sabrina Baumann** and **Tanja Turner** for their attitude, life advice, shielding me from bureaucracy, and wonderfully organized events and aperos at IfA. I thank **Rita Klute** for her support and perfect organization of events in the LAS group.

I am glad I have friends who always were close by and willing to help and inspire: **Dr. Sergey Dovgal**, and **Dr. Natasha Kharchenko**, **Misha Karasikov**, **Oleg Ponomarev**, **Tanya** and **Kostya**, **Ira**, and **Borya**, and **Natasha**. A special thanks to **Anton Obukhov** for proofreading parts of this dissertation. Thank you **Misha** for supporting me in any mood, for your endless motivation and energy, and for being my best friend and husband.

I want to express great gratitude to my family. Дорогие **мама**, **папа**, **Гузаль** и **Данил**, спасибо вам за всю доброту и тепло, которые я имела счатье испытать будучи частью нашей большой и дружной семьи.

Finally, we thank the support of Swiss National Science Foundation, under the grant SNSF 200021_172781, the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40 180545, and European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No 815943.

Abstract

Optimizing noisy functions online, when evaluating the objective requires experiments on a deployed system, is a crucial task arising in manufacturing, robotics, medical trials, and other domains. Often, in such systems constraints on safe inputs are unknown ahead of time, and we only obtain noisy information indicating how close we are to violating the constraints. However, safety must be guaranteed at all times, not only for the final output of the algorithm. Indeed, in many applications, one wants to perform learning at least partially in the real world. For example, in robotics, after pre-training a policy on simulations, one would like to employ it in the real world and keep improving it directly using robot-environment interactions. Keeping the updates safe is crucial not to harm the platform. In personalized medicine, one would like to apply a therapy based on clinical trials to actual patients, without harming the patients. Initially, a doctor would prescribe a particular patient very conservative dosages of new medicine. By monitoring the patient, she can observe how the patient tolerates the medicine and try to make gentle iterative changes to adjust the dosage for this patient safely. Our idea is similar – we propose optimization methods that iteratively update the decision variables using interactions with the environment in a safe way.

The main goal of this thesis is to demonstrate that we can address safe learning problems using feasible constrained optimization techniques. Using such simple techniques allows for addressing high dimensional problems, in contrast to more complicated approaches such as Bayesian Optimization.

We propose two general approaches. The first approach addresses safe learning under uncertain *linear* constraints. It is based on the Frank-Wolfe method combined with the robust optimization technique, allowing us to guarantee safety under uncertainty. We prove its convergence for convex problems and guarantee the safety of all the measurements taken during the learning with high probability. The second approach is more general and allows for addressing the optimization problems with *non-linear* constraints. Its main idea is to use the logarithmic barriers to address safety. To minimize the log barrier subproblem, it uses simple and powerful stochastic gradient descent (SGD) with a carefully chosen adaptive step size. We provide the analysis of the convergence rate of this method for non-convex, convex, and strongly-convex problems. Furthermore, we analyze the sample complexity of our method separately, given the first-order stochastic oracle and zeroth-order noisy oracle. Additionally to the analysis of smooth problems, we provide the extension addressing the non-smooth problems and analyze its sample complexity.

We compare our methods on synthetic problems with existing baselines and show their performance in applications such as manufacturing, control, and reinforcement learning.

Kurzfassung

Die Online-Optimierung verrauschter Funktionen, deren Evaluierung Experimente an einem eingesetzten System erfordert, ist eine wichtige Aufgabe in der Fertigung, der Robotik, bei klinischen Versuchen und in anderen Bereichen. In solchen Systemen sind die Bedingungen für sichere Inputs häufig a priori unbekannt, und man erhält lediglich verrauschte Informationen, wie nahe man an einer Verletzung der Bedingungen ist. Die Sicherheit muss jedoch zu jedem Zeitpunkt gewährleistet sein und nicht nur für den finalen Output des Algorithmus. In der Tat ist es bei vielen Anwendungen wünschenswert, das Lernen zumindest teilweise in der realen Welt durchzuführen. In der Robotik, zum Beispiel, würde man gerne nach dem Pre-training einer Policy durch Simulationen, diese in der realen Welt einsetzen und sie durch Interaktionen zwischen dem Roboter und der Umgebung weiter verbessern. Dabei ist die Sicherheit jedes Updates essenziell, um die Plattform nicht zu beschädigen. Ein weiteres Beispiel ist die personalisierte Medizin, in der man basierend auf klinischen Studien mögliche Therapieformen an Patienten testen möchte, ohne diese zu gefährden. Zu Beginn wird der behandelnde Arzt dem Patienten eine sehr geringe Dosierung des neuen Medikaments verabreichen und wird dann, anhand der beobachteten Reaktion des Patienten auf das Medikament, die Dosierung schrittweise und sicher anpassen. Unsere Idee funktioniert analog, dabei schlagen wir Optimierungsmethoden vor, welche die Wechselwirkungen mit der Umwelt nutzen, um die Entscheidungsvariablen auf sichere Weise iterativ zu aktualisieren.

Das Ziel dieser Arbeit ist es, zu zeigen, dass wir sichere Lernprobleme mit Hilfe zulässiger Optimierungstechniken für Optimierungsproblem mit Nebenbedingungen lösen können. Im Gegensatz zu komplizierteren Ansätzen, wie z.B. der Bayes'schen Optimierung, können mit solchen einfachen Techniken auch hochdimensionale Probleme gelöst werden.

Wir schlagen zwei allgemeine Ansätze vor. Der erste Ansatz befasst sich mit sicherem Lernen unter unsicheren linearen Randbedingungen. Er basiert auf der Frank-Wolfe-Methode, die zusammen mit robusten Optimierungstechniken, Sicherheit unter Ungewissheit garantiert. Wir beweisen einerseits die Konvergenz unseres Ansatzes für konvexe Probleme und zeigen andererseits, dass die Sicherheit der während des Lernens durchgeführten Messungen mit hoher Wahrscheinlichkeit garantiert ist. Der zweite Ansatz ist allgemeiner und ermöglicht die Lösung von Optimierungsproblemen mit nichtlinearen Nebenbedingungen. Die Hauptidee besteht darin, logarithmische Barrieren zu verwenden, um die Sicherheit zu gewährleisten. Um das logarithmische Barrieren-Subproblem zu minimieren, verwendet man ein einfaches und doch leistungsfähiges stochastic gradient descent Verfahren (SGD) mit einer sorgfältig gewählten, adaptiven Schrittgröße. Diese Arbeit analysiert die Konvergenzrate dieser Methode für nicht-konvexe, konvexe und starkkonvexe Probleme. Des Weiteren wird in dieser Arbeit die Stichprobenkomplexität (sample complexity) der vorgeschlagenen Methode separat für stochastische Orakel erster Ordnung und das verrauschte Orakel nullter Ordnung untersucht. Zusätzlich zu der Analyse glatter Probleme, bietet diese Arbeit eine Erweiterung für nicht-glatte Probleme und analysiert auch deren Stichprobenkomplexität. Schlussendlich vergleichen wir unsere Methoden auf synthetischen Problemen mit existierenden Baselines und zeigen ihre Stärken in verschieden Anwendungen wie der Fertigung, Steuerung und Reinforcement Learning

Contents

Acronyms and notation

For the sake of clarity, some of the acronyms above will be defined again when they appear in the dissertation for the first time.

Mathematical symbols

x	vector
X	matrix
$\mathbb{R}^d$	real coordinate space of dimension d
$\mathbb{R}^d_+$	non-negative orthant of dimension d
$[m]$	set of all positive integers up to integer m
$\|\cdot\|$	ℓ_2-norm (Euclidean norm)
$\mathcal{X}$	feasibility set
$Int(\mathcal{X})$	interior of the set $\mathcal{X}$
D	diameter of the set $\mathcal{X}$
$\Pi_{\mathcal{X}}$	Euclidean projection onto $\mathcal{X}$
$\mathcal{S}^d$	unit sphere in $\mathbb{R}^d$
$\mathcal{B}^d$	unit ball in $\mathbb{R}^d$
$\mathcal{S}^d(x_0, r)$	Euclidean sphere in $\mathbb{R}^d$ centered at $x_0 \in \mathbb{R}^d$ and with radius $r \in \mathbb{R}_+$
$\mathcal{B}^d(x_0, r)$	Euclidean ball in $\mathbb{R}^d$ centered at $x_0 \in \mathbb{R}^d$ and with radius $r \in \mathbb{R}_+$
$[\cdot]_+$	positive cut, equal to $(\cdot)$ when it is positive, and equal to 0 otherwise
$f(x) = O(g(x))$	big O notation
$f(x) = \tilde{O}(g(x))$	big O up to a multiplicative logarithmic factor
$f(x) = \Omega(g(x))$	big Ω notation
$\mathcal{O}(\cdot)$	oracle
$\mathbb{P}\{\mathcal{A}\}$	probability of event $\mathcal{A}$
$\mathbb{E}_s[\cdot]$	expectation calculated over the distribution of the random variable s
$\mathcal{D}$	data set
ξ	a vector of random variables
ε	accuracy
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean vector μ and covariance matrix Σ
$\mathcal{L}(\cdot, \cdot)$	Lagrangian function
λ	Lagrange dual vector

Acronyms / Abbreviations

FW	Frank-Wolfe method
RO	Robust optimization
IPM	Interior point methods
SGD	Stochastic gradient descent
LB-SGD	Log barriers SGD
RL	Reinforcement learning
RKHS	Reproducing kernel Hilbert space
GP	Gaussian process
LP	Linear programming
NLP	Non-linear programming
QP	Quadratic programming
BO	Bayesian optimization
MDP	Markov decision process
CMDP	Constrained Markov decision process
POMDP	Partially observable Markov decision process
NN	Neural network
LQR	Linear quadratic regulator

CHAPTER 1

Introduction

1.1 Motivation and goals

Many optimization tasks in robotics, health sciences, and finance require minimizing a loss function under uncertainties. Most existing stochastic and online optimization approaches propose to address these tasks assuming that the constraints of the corresponding optimization problems are known. These approaches, however, are unacceptable in cases in which the feasible set is itself *unknown* and is *learned online*. Optimizing a loss function under such a partially revealed feasible set model is further challenged by the fact that exploration can be made only inside the feasible set due to safety reasons. Hence, one needs to carefully choose actions to ensure the feasibility of each iterate with high probability while learning the optimal solution. In the machine learning community, this problem is known as *safe learning*. Our goal is to include safety in existing learning techniques.

Motivation Safe learning is receiving increasing attention due to the increasingly widespread deployment of machine learning in safety-critical tasks. An example arises in personalized medicine, where physicians may choose from a large set of therapies. The effects of different therapies on the patient are initially unknown and can only be determined through clinical trials. Free exploration, however, is not possible since some therapies might cause discomfort or even physical harm [Sui+15a]. Similar challenges arise in designing control algorithms for robots, which have to navigate unexplored terrains or interact with humans [CKK+96; KS96]. In these scenarios, robots need to learn the best tuning for their controllers or optimize their trajectories based on risky experimental interactions with partially unknown environments. Indeed, to be able to perform safe learning, one requires to start from a known safe initial point. This is a classical assumption for the works addressing safe learning. Without this assumption, we cannot guarantee safety even for the initial point. Although this is quite a strong assumption, such tasks also appear in practice. We consider a couple of examples below.

Example: Autonomous Driving The first example is a control policy learning problem for an autonomous driving car [INF18; Wen+20; Fai+19; SAR18; DR18; Kat+15].

Typically, this policy has to be trained in a particular city where the learner has the complete infrastructure for that, let us assume this city is Zürich. The car collects a huge amount of data from this city and trains a reliable policy π_1. The manufacturers make the policy π_1 conservative enough to be robust and preserve safety in all other cities in the country. However, they make it maximally efficient for Zürich. Then, assume that one wants to drive this car in Lausanne. The policy π_1 is still safe *by design*. But it is not optimal anymore since the conditions and environment have changed [Zha+21]. For instance, due to the increased hilliness overtaking drivers in Lausanne may be harder for π_1; therefore, reaching the destination takes a longer time. Then, one wants to train a new policy to be more efficient in Lausanne. There are two ways to do this. One way is to repeat everything that was done in Zürich from scratch: build infrastructure, collect a massive amount of data, and then train the policy. The second option would be to start running the car in Lausanne, and update the policy online, iteratively adapting it to the new environment. The second option is obviously much cheaper. However, it requires very strict safety guarantees not only at the end of learning but also during the adaptation process. Therefore, it becomes a pure *safe learning* problem. Starting from a safe policy π_1, one wants to fine-tune it to the new environment in order to improve its performance. However, due to real-world interactions, it is crucial to keep the learning safe. The same idea can be used for adapting a policy pre-trained on a simulator for the real world, the so-called *sim-to-real* problem. See Akhauri, Zheng, and Lin [AZL20] for an example of using transfer learning for autonomous driving cars, or Nowruzi, Kapoor, Kolhatkar, Hassanat, Laganiere, and Rebut [Now+19] for an example of using video-games simulations to train autonomous driving cars.

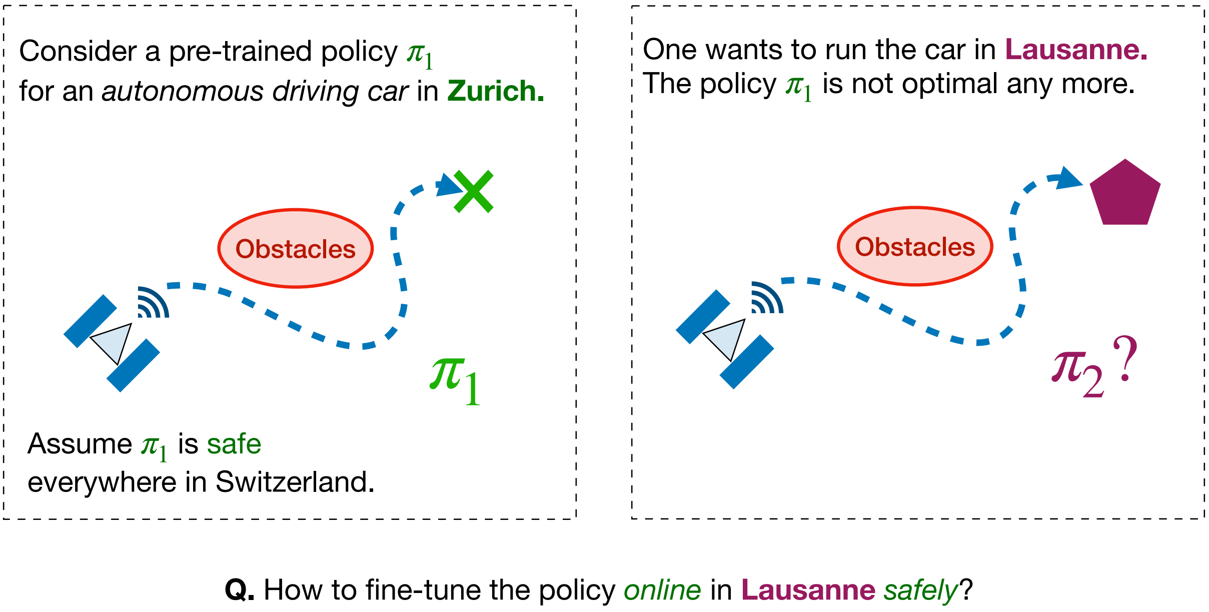

Consider a pre-trained policy π_1
for an *autonomous driving car* in **Zurich.**
Obstacles
π_1
Assume π_1 is safe
everywhere in Switzerland.
One wants to run the car in **Lausanne.**
The policy π_1 is not optimal any more.
Obstacles
π_2?
Q. How to fine-tune the policy *online* in **Lausanne** *safely*?

Figure 1.1: Illustration of an autonomous driving example

Example: Manufacturing Another interesting example is the parameters tuning in manufacturing. Rapid technological development allows for fine-tuning the parameters of machines to improve their performance online by collecting data. Imagine one needs to tune the parameters of an industrial machine, say, a cutting machine, such as the turning speed, angle, etc., in order to minimize the costs of production. Typically, in the industry, such parameters are tuned manually by experts. In fact, it happens by trial and error. Moreover, the optimization must be done in such a way that the parameters do not violate safety constraints such as upper and lower limits of energy consumption [NM10], or power constraint [Rat+21], or product quality [Mai+18]. Therefore, during the production process, one could improve the performance of the machine using feedback. Choosing the parameters during the learning process that allow safety constraint violations can lead to breaking the machine or producing products with unsatisfactory quality. We want all the updates during the learning to guarantee a user-defined minimum performance. Therefore, one should use safe online learning techniques that ensure feasible updates during the learning.

Goals There are works addressing safe learning in the past. A significant part of these works is based on the Bayesian optimization (BO) [Sui+15b; BKS16]. The authors address the safe optimization under the assumption that the objective and constraints have a bounded reproducing kernel Hilbert space (RKHS) norm. Based on the obtained measurements, they build a model of the objective and constraints based on Gaussian processes (GP) that allows one to determine the safe region for choosing the next query point. The Bayesian approach has a significant disadvantage since it does not scale for medium-to-high dimensions due to the curse of dimensionality. Some extensions try to overcome this disadvantage [Kir+19]; however, they are not universal yet. This motivates our work of developing efficient, safe learning algorithms applicable to higher dimensions. For high-dimensional optimization, gradient-based optimization methods have shown a good performance and are widely used in tasks such as neural network (NN) parameters optimization. Stochastic first- and zeroth-order methods are already widely explored for constrained non-convex, convex, and strongly convex problems. In this dissertation, we want to use the power of simplicity and the cheapness of gradient-based optimization methods to address the safe learning task.

- As a first step, our goal is to address the simpler problem of safe learning under the assumption that the constraints are linear. In this case, one could use the local measurements of the constraints for linear regression to estimate a model of the constraint set and to use it for our learning.

- The second goal is to propose a more general gradient-based method applicable to non-linear constraints. We also want to investigate the advantages and limitations of such an approach.

- The final goal is to demonstrate that the proposed techniques can be applied to real learning problems such as control or safe reinforcement learning.

1.2 Outline and contributions

In the following, we present a brief summary and outline of each chapter.

1.2.1 Chapter 2

In this chapter, we provide the necessary background related to our work. In particular, in Section 2.1 we define the important notions related to the first- and zeroth-order optimization that we use in the current dissertation. Then, in Section 2.2 we provide the overview of existing methods addressing constrained optimization and safe learning.

1.2.2 Chapter 3

The chapter is based on our work by Usmanova, Krause, and Kamgarpour [UKK19]. In this chapter, we address the safe learning problem with polytopic constraints. We propose a safe robust algorithm based on the Frank-Wolfe method. In particular, in Section 3.1 we formulate the safe learning problem with polytopic constraints. Then, in Section 3.2 we define our Safe Frank-Wolfe (SFW) algorithm. In Section 3.3 we analyze the safety of our method, and then, in Section 3.4 we provide the convergence analysis. Finally, in Section 3.5 we empirically demonstrate the performance of our algorithm on simulations. In Appendix A we provide all the necessary proofs.

Overview of contributions

- We propose a novel algorithm for safe active learning, given a smooth convex objective and a set of unknown linear constraints with noisy oracle information. The core idea of our algorithm is to combine a first-order feasible optimization approach with a robust optimization technique. Specifically, our algorithm is based on the Frank-Wolfe (FW) method [FW56]. At each iteration, it solves an *uncertain* linear program based on estimates of the constraints and uses this solution to define the step direction. The safety of the iterates is ensured with high probability by iteratively refining the confidence set of the unknown parameters.

- Given a confidence level $1-\delta$ and accuracy ε, we prove that after $\tilde{O}\left(\frac{1}{\varepsilon}\right)$ iterations (one objective gradient measurement per iteration) and $\tilde{O}\left(\frac{d^2 \ln 1/\delta}{\varepsilon^2}\right)$ constraints measurements in total, the final point is an ε-accurate solution with probability $1-\delta$ (Theorem 2). By $\tilde{O}(\cdot)$ we denote $O(\cdot)$ up to a logarithmic multiplicative factor.

- Furthermore, we ensure feasibility for the trajectory of the iterates with probability at least $1-\delta$ (Theorem 1). While in this chapter we mainly focus on exact first-order oracles for the objective function, we discuss extensions to stochastic oracles in Section 3.4.

- We evaluate the performance of the proposed algorithm numerically in Section 3.5 and compare its performance with a one-shot robust optimization approach.

1.2.3 Chapter 4

In Chapter 4 we address general smooth non-linear objective and constraints. This chapter is mainly based on our paper by Usmanova, As, Kamgarpour, and Krause [Usm+22] and additionally uses the results obtained in Usmanova, Krause, and Kamgarpour [UKK20]. We build a method based on the logarithmic barriers, and propose to apply Stochastic Gradient Descent (SGD) to minimize it, by carefully choosing adaptive step size. We discuss advantages and limitations of this approach, and analyze it convergence rate and safety for various types of smooth problems such as non-convex, convex, and strongly-convex. In particular, in Section 4.1 we define the problem and the oracle information. In Section 4.2 we introduce our general approach to the safe learning using logarithmic barriers. In this section, we specify how to choose the step size and direction used in our method, formulate a basic algorithm for smooth problems with first-order oracle, and prove its safety. In Section 4.3 we focus on various method variants and their convergence rates for different types of smooth problems. In particular, we analyze the non-convex, convex, and strongly-convex settings. In Section 4.4 we extend our results to zeroth-order information setting, and finally, in Section 4.5 we extend our results to the non-smooth optimization setting, which we address using the randomized sampling technique. Section 4.5 is not the part of the paper [Usm+22] but uses some results of [UKK20]. In Section 4.6 we empirically compare our work on simulations with other safe learning approaches. In Appendix B we provide the necessary proofs.

Overview of contributions

- We propose a unified approach for safe learning given zeroth-order or first-order stochastic oracle. We prove that our approach generates feasible iterations with high probability and converges to a stationary point (or to the optimum in the convex case). Each iteration of the proposed method is computationally cheap and does not require solving any subproblems such as those required for Frank-Wolfe (LP subproblems) or BO-based algorithms (NLP subproblems).

- We derive the convergence rate of our algorithm for the stochastic non-convex, convex, and strongly-convex problems. We prove the convergence despite the non-smoothness of the log barrier and the increasingly high variance of the log barrier gradient estimator.

- We address the zeroth-order information case, when one can measure only the noisy values, and has no access even to stochastic gradients of the objective and constraint functions. We extend the above approaches to address this issue by using the randomized smoothing technique and finite differences to estimate the gradients. This technique also allows us to address non-smooth problems.

- We empirically show on simulations that our method can scale to problems with high dimensions in which previous methods fail.

1.2.4 Chapter 5

In Chapter 5 we discuss applications of our safe learning methods (in particular, of the log barrier approach) to real-world problems. In particular, we apply the logarithmic barrier approach to the parameters tuning problem in manufacturing, to the dynamics controller optimization in linear-quadratic regulator problem (LQR), and to the safe model-based reinforcement learning (RL). In Section 5.1 we apply our method for tuning parameters in the cutting machine showcase. In Section 5.2 we demonstrate the performance of our method in an application of controller learning in LQR problem. These sections use the results of our past work Usmanova, Krause, and Kamgarpour [UKK20]. In Section 5.3 we discuss the application of the log barrier approach to a high-dimensional problem of safe model-based reinforcement learning (RL) problem. This section is based on Usmanova, As, Kamgarpour, and Krause [Usm+22], which in turn also strongly uses the results of As, Usmanova, Curi, and Krause [As+22]. The code in this subsection was implemented by Yarden As, my contribution was in proposing the log barriers approach for this problem, and in designing the experiment in such a way that it is suitable for the safe learning setup. We provide the necessary background on model-based RL, describe our experiments using log barriers approach, and demonstrate its performance. In Appendix C we provide some additional materials for this chapter.

1.3 Publications

This thesis contains a selected collection of results derived during the author's studies as a Ph.D. candidate. The corresponding articles on which this thesis is based are listed below.

[UKK19] "*Safe Convex Learning Under Uncertain Constraints*", **I. Usmanova**, A. Krause, M. Kamgarpour, International Conference on Artificial Intelligence and Statistics (AISTATS), 2019;

[UKK20] "*Safe non-smooth black-box optimization with application to policy search*", **I. Usmanova**, A. Krause, M. Kamgarpour, 2nd Annual Conference on Learning for Dynamics and Control (L4DC), 2020 ;

[As+22] "*Constrained Policy Optimization via Bayesian World Models*", Y. As, **I. Usmanova**, S. Curi, A. Krause, International Conference on Learning Representations (ICLR), 2022;

[Usm+22] "*Log Barriers for Safe Black-box Optimization with Application to Reinforcement Learning*", **I. Usmanova**, Y. As, M. Kamgarpour, A. Krause, Under review at Journal of Machine Learning Research (JMLR), 2022;

1.3.1 Other publications

The following papers were published by the author during her doctoral studies, but are not included in the thesis.

[Usm+21] "*Fast Projection Onto Convex Smooth Constraints*", **I. Usmanova**, M. Kamgarpour, A. Krause, K.Y.Levy, International Conference on Machine Learning (ICML), 2021;

[Mak+21] "*Risk-averse Heteroscedastic Bayesian Optimization*", A. Makarova, **I. Usmanova**, I. Bogunovich, A. Krause, Conference on Neural Information Processing Systems (NeurIPS), 2021;

The first reference concerns the fast projection technique for solving high dimensional projection problems onto the intersection of a few convex smooth constraints. The second reference is on risk-averse Bayesian optimization in the heteroscedastic setting.

CHAPTER 2

Background and Related Work

This dissertation uses various basic constrained optimization techniques. In this chapter we provide the necessary mathematical background for our work, and the review of the literature related to the safe learning task and to ideas that we use.

2.1 Preliminaries

Before moving to the main part of the thesis, we first define the important notions that we use in our work. We use the notations and definitions in this chapter throughout the thesis.

Optimization problem Throughout this thesis, we consider optimization problems in the form

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \ & f^0(x) \\ \text{s.t.} \ & f^i(x) \leq 0, i \in [m] \end{aligned} \tag{P}$$

where *s.t.* stands for *subject to*, and f^0 is the objective function. The constraint set $\mathcal{X}$ here is given by m constraint functions $\mathcal{X} := \{x \in \mathbb{R}^d : f^i(x) \leq 0, \forall i \in [m]\}$. Alternatively, the problem can also be formulated as $\min_{x \in \mathcal{X}} f^0(x)$.

Duality The Lagrangian function of the constrained problem (P) is defined as follows

$$\mathcal{L}(x, \lambda) := f^0(x) + \sum_{i=1}^{m} \lambda_i f^i(x).$$

Hereby, $\lambda \in \mathbb{R}^m$ is the vector of dual variables. Then the problem (P) is equivalent to:

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \in \mathbb{R}^m_+} \mathcal{L}(x, \lambda).$$

The dual problem is then defined by changing the order of min and max:

$$\max_{\lambda \in \mathbb{R}_+^m} g(\lambda) = \max_{\lambda \in \mathbb{R}_+^m} \min_{x \in \mathbb{R}^d} [f^0(x) + \sum_{i=1}^m \lambda_i f^i(x)]. \tag{D}$$

Convexity The function $f(\cdot)$ is called to be *convex* on $\mathcal{X}$ if for any $x, y \in \mathcal{X}$ we have $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ for any $\alpha \in [0,1]$. If the function is differentiable, it is equivalent to

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle.$$

The set $\mathcal{X}$ is called convex if for any $x, y \in \mathcal{X}$ we have

$$\alpha x + (1-\alpha) y \in \mathcal{X}.$$

The set $\mathcal{X}$ defined by the convex constraint functions is also convex. We call the problem convex if both the objective $f^0(x)$ and the constrained set $\mathcal{X}$ are convex.

For convex problems, the optimal values of the primal problem (P) and the dual problem (D) coincide.

Smoothness, Lipschitz continuity, and strong convexity A function $f(x)$ is called *L-Lipschitz continuous* on $\mathcal{X}$ if

$$|f(x) - f(y)| \leq L\|x - y\|, \, \forall x, y \in \mathcal{X}. \tag{2.1}$$

It is called *M-smooth* on $\mathcal{X}$ if the gradients $\nabla f(x)$ are M-Lipschitz continuous, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|, \, \forall x, y \in \mathcal{X}. \tag{2.2}$$

or equivalently,

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{M}{2}\|x - y\|^2.$$

The function is called μ-strongly-convex on $\mathcal{X}$, if

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|x - y\|^2, \, \forall x, y \in \mathcal{X}.$$

Oracle model and uncertainty The information is usually provided by measurements at the requested points. Exact zeroth-order oracle provides the value $f(x)$ at the requested point x. Exact first-order oracle for the differentiable function f provides the gradient $\nabla f(x)$ at the requested point x. But typically, in the applications we consider, the information available to the learner is noisy. For example, one can only observe perturbed gradients and values of $f^i, \forall i = 0, \ldots, m$ at the requested points x_t. In particular, we assume that the oracles are corrupted by an additive sub-Gaussian noise. A random variable ξ is called zero-mean σ^2-*sub-Gaussian* if $\forall \lambda \in \mathbb{R}\ \mathbb{E}\left[e^{\lambda\xi}\right] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right)$, which implies that $\mathrm{Var}\left[\xi\right] \leq \sigma^2$ (this can be shown using Tailor expansion). We formally define the first-order stochastic oracle and the zeroth-order stochastic oracle below.

Zeroth-order stochastic oracle When we consider a zeroth-order oracle, we assume we have access to a *one-point stochastic zeroth-order oracle*, defined as follows. For any $i \in \{0, \dots, m\}$ this oracle provides noisy function evaluations at the requested point x_j:

$$F^i(x_j, \xi^i_j) = f^i(x_j) + \xi^i_j, \tag{2.3}$$

where ξ^i_j is a zero-mean σ_i^2-sub-Gaussian noise. We assume that noise values ξ^i_j may differ over iterations j and indices i even for the close points, i.e., we cannot access the evaluations of f^i with the same noise by two different queries: $\xi^i_j \neq \xi^i_{j+1}$ for any $F^i(x_j, \xi^i_j)$ and $F^i(x_{j+1}, \xi^i_{j+1})$ even if $x_j = x_{j+1}$. This is in contrast to the *two-point stochastic zeroth-order oracle* that allows to have evaluations with the same noise vector at two different points, which is a significantly stronger assumption [Duc+15]. Also, we assume that the measurements taken several times at the same point are i.i.d. random variables.

First-order stochastic oracle In this case, we consider access to the first-order stochastic oracle for every $f^i(x)$, providing the pair of value and gradient stochastic measurements:

$$\mathcal{O}(f^i, x, \xi) = (F^i(x, \xi), G^i(x, \xi)). \tag{2.4}$$

Note that the formulation allows (but does not require) that $F^i(x, \xi)$ and $G^i(x, \xi)$ are correlated. In particular, this formulation allows to define the vector of $\xi = \{(\xi^i_0, \xi^i_1)\}_{i=0,\dots,m}$ such that each $F^i(x, \xi) = F^i(x, \xi^i_0)$ and $G^i(x, \xi) = F^i(x, \xi^i_1)$. In this formulation, $\{(\xi^i_0, \xi^i_1)\}_{i=0,\dots,m}$ can be either correlated or independent of each-other. The parts of the oracle are given as follows:

1) **Stochastic value** $F^i(x, \xi)$**.** We assume $F^i(x, \xi)$ is unbiased

$$\mathbb{E}[F^i(x, \xi)] = f^i(x),$$

and sub-Gaussian with variance bounded by σ_i^2, that is, for any $\delta \in (0, 1)$

$$\mathbb{P}\left\{|F^i(x, \xi) - f^i(x)| \leq \sigma_i \sqrt{\ln \frac{1}{\delta}}\right\} \geq 1 - \delta,\ i \in \{0, \dots, m\}.$$

2) **Stochastic gradient** $G^i(x, \xi)$**.** We assume that its bias is bounded by

$$\|\mathbb{E}G^i(x, \xi) - \nabla f^i(x)\| \leq \hat{b}_i,$$

where $\hat{b}_i \geq 0$, and it is sub-Gaussian with the variance such that $\mathbb{E}[\|G^i(x,\xi) - \mathbb{E}G^i(x,\xi)\|^2] \leq \hat{\sigma}_i^2$ for some $\hat{\sigma}_i \geq 0$.

Optimality criteria The goal of an optimization method is to solve problem (P) up to some level of accuracy $\varepsilon > 0$, provided with an information oracle.

In *convex* case we search for an ε-approximate solution $\hat{x}$ such that

$$|f^0(\hat{x}) - \min_{x \in \mathcal{X}} f^0(x)| \leq \varepsilon. \tag{2.5}$$

If the problem is *non-convex*, finding the global optimum might be an NP-hard task; therefore, we seek for a *stationary point*. In particular, for differentiable problems, we search for a stationary point defined by the Karush-Kuhn-Tucker (KKT) conditions [KT51]

$$\lambda_i, -f^i(x) \geq 0, \ \forall i \in [m] \tag{KKT.1}$$
$$\lambda_i(-f^i(x)) = 0, \ \forall i \in [m] \tag{KKT.2}$$
$$\|\nabla_x \mathcal{L}(x,\lambda)\| = 0. \tag{KKT.3}$$

To measure the approximation in the non-convex differentiable case we define the *ε-approximate KKT point* **(ε-KKT)**. Specifically, for $\varepsilon > 0$ and a pair (x,λ), such point satisfies the following conditions:

$$\lambda_i, -f^i(x) \geq 0, \ \forall i \in [m] \tag{ε-KKT.1}$$
$$\lambda_i(-f^i(x)) \leq \varepsilon, \ \forall i \in [m] \tag{ε-KKT.2}$$
$$\|\nabla_x \mathcal{L}(x,\lambda)\| \leq \varepsilon. \tag{ε-KKT.3}$$

This definition is similar to an unscaled approximate KKT point in Hinder and Ye [HY19] with the only difference that we require our point to be feasible. In some works, e.g., [BG18], another stationarity condition is used for constrained non-convex optimization given that set $\mathcal{X}$ is convex and closed:

$$\langle \nabla f^0(x), x - u \rangle \leq \varepsilon, \forall u \in \mathcal{X}. \tag{2.6}$$

In this dissertation, we do not make such an assumption of convexity of $\mathcal{X}$ for the non-convex problems. However, note that when $\varepsilon \to 0$, for convex and closed set $\mathcal{X}$ one can show that x is a KKT point if and only if x satisfies Eq. (2.6) with $\varepsilon = 0$ (see for example Zhao and Gordon [ZG17] Theorem 1).

We say we solve the *safe learning* task if we require for all trajectory $\{x_t\}_{t\in[T]}$ generated by the algorithm during the learning, to satisfy $\{x_t\}_{t\in[T]} \in \mathcal{X}$ with high probability $1-\delta$ for some confidence level $\delta \in (0,1)$.

Sample complexity To measure the efficiency of an optimization algorithm, we use the notion of *sample complexity*, also known as oracle complexity introduced by Nemirovsky and Yudin [NY85]. That is, how many oracle calls $N(\varepsilon)$ are needed for the algorithm to reach accuracy ε. Another notion is *computational complexity*, which denotes the number of arithmetical operations required by the method to achieve accuracy ε.

2.2 Literature review

Next, we provide an overview of the existing methods in the literature related to the safe learning problem. Consider the following optimization problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^d} f^0(x) \\ &\text{ s.t. } f^i(x) \leq 0, i \in [m] \end{aligned} \tag{P}$$

under the partial information provided by zeroth-order or first-order oracle of f^i for all $i \in [m]$, for example, as defined in Eq. (2.4) or Eq. (2.3). Optimizing a loss function under such partially revealed information is further challenged by the fact that exploration can be made only inside the feasible set due to safety reasons. That is, one needs to carefully choose actions to ensure the feasibility of each iterate with high probability while learning the optimal solution. In particular, the key requirement is *safe learning*: during the optimization procedure one has to keep all the iterates x_t inside the feasibility set $\mathcal{X} := \{x \in \mathbb{R}^d : f^i(x) \leq 0, \ \forall i \in [m]\}$ with high probability $1-\delta$ for some $\delta \in (0,1)$. Note that depending on the information available, one can consider a safe first-order optimization problem or a safe zeroth-order problem. For instance, when one has access to a stochastic model describing the objective and constraints, one can think of the first-order problem. Alternatively, when the learner only can measure the noisy values of the objective and constraints using real-world interactions, we talk about the zeroth-order optimization problem.

We first provide some background on the classical stochastic first-order constrained optimization methods and then on the relevant zeroth-order optimization methods. Optimization methods can be categorized into two classes. The first class is the methods assuming *known* constraints requiring exact global information of the constraint set $\mathcal{X}$, allowing, for example, to project onto $\mathcal{X}$, or to solve a linear sub-problem subject to $\mathcal{X}$. The second class is those assuming *unknown* constraints that require only local information such as gradient or value measurements of $f^i(x)$. This information also can be either stochastic or exact. Moreover, the methods can be *feasible* or *infeasible*. By feasible optimization approaches, we mean any constrained optimization methods that generate a feasible optimization trajectory $x_t \in \mathcal{X} \ \forall t > 0$. In contrast, infeasible methods only care about the feasibility of the final output and do not guarantee feasibility during the optimization. Feasible methods are especially interesting for us, even in the case of known constraints, since they can be potentially extended to the harder setting of unknown constraints.

In the following sections, we first bring the existing relevant methods in first-order stochastic optimization, assuming both known or unknown constraints. Next, we discuss the existing methods addressing zeroth-order optimization. This includes methods assuming known constraints and, finally, those feasible methods addressing unknown, uncertain constraints, which corresponds exactly to the safe learning setting. We finally describe the existing safe learning methods to address the zeroth-order problems together with our approaches.

2.2.1 First-order constrained stochastic optimization

Known constraint set $\mathcal{X}$ There are many first-order optimization algorithms that ensure the feasibility of the iterates, assuming that the constraint set is known. The most basic ones are the projected gradient descent (PGD)[BV04] and Frank-Wolfe (FW) [FW56] (also known as conditional gradient).

Projected gradient descent: Robbins and Monro, in their pioneering work [RM51] in 1951, proposed a stochastic approximation (SA) method that mimics the simplest gradient descent approach: $x_{t+1} \leftarrow x_t - \gamma_t \nabla f(x_t)$ by using noisy gradient measurements instead of the gradient: $x_{t+1} \leftarrow x_t - \gamma_t \nabla F(x_t, \xi_t)$. In the above, γ_t is some step size (sometimes, in machine learning applications, it is called a learning rate). The basic version of this method is called stochastic gradient descent (SGD). It achieves in general the optimal convergence rate $O(\frac{1}{\varepsilon^2})$. Since then, SA methods have become widely used in stochastic optimization and were further developed in [Nem+09; Jud+13; Lan20; DSS21]. These types of methods are also used for non-convex problems, e.g., see Ghadimi and Lan [GL13]. To address the constraints, this class of methods proposes to use projections:

$$x_{t+1} \leftarrow \Pi_{\mathcal{X}}(x_t - \gamma_t \nabla F(x_t, \xi_t)).$$

This approach is feasible during learning; however, it requires a projection oracle on the set $\mathcal{X}$ providing the result of operation $\Pi_{\mathcal{X}}(x)$. For some sets $\mathcal{X}$ projection operation $\Pi_{\mathcal{X}}(x)$ is very simple, e.g., projection on to the unit ball is equivalent to the normalization: $\Pi_{\mathcal{S}^d}(x) = \frac{x}{\|x\|}$. Whereas for general sets, it might not have an analytical form and can be considered as a separate complicated subproblem.

Frank-Wolfe method: A projection-free way of solving constrained stochastic problems is the Frank-Wolfe approach. The Frank-Wolfe method, also known as the conditional gradient method, was proposed by Frank and Wolfe in 1956 [FW56]. The procedure is very simple: at every iteration, t, one has to minimize the linear gradient subject to the constraint set $\mathcal{X}$.

$$s_t \leftarrow \arg\min_{s \in \mathcal{X}} \langle s, \nabla f^i(x_t) \rangle$$
$$x_{t+1} \leftarrow (1 - \gamma_t) x_t + \gamma_t s_t, \text{for } \gamma_t = \frac{2}{t+2}$$

Recently, this method has become popular again. The analysis of stochastic version of it was analyzed in [Jag13], [LJ13]. This approach is also feasible since all the updates, by definition, lie within the set $\mathcal{X}$, but again, it can only be employed given the LP oracle subject to $\mathcal{X}$.

These classical projected SGD and FW methods require exact knowledge of the constraints or at least a projection oracle or an exact linear programming (LP) oracle with respect to the constraints. That means, given a too general structure of the constraints, they cannot be applied directly to the safe learning problem. However, by assuming a particular class of constraints, such as linear constraints, we can estimate the whole feasibility set using regression. This allows using ideas similar to Frank-Wolfe for safe learning, which we demonstrate later in this thesis.

Unknown constraint set $\mathcal{X}$ Now we consider other classes of algorithms that address constraints given in a general form $\mathcal{X} = \{x \in \mathbb{R}^d | f^i(x) \leq 0 \; \forall i \in [m]\}$ by using only the local information such as values and gradients.

Penalty methods: A recent line of work addresses uncertain constraints in online stochastic optimization [YNW17; YN16]. The work is based on infeasible penalty methods and thus does not provide guarantees on constraint violation at each iteration. The idea is to replace the constraint optimization problem with the sequence of its unconstrained approximation $\min_{x\in\mathbb{R}^d} f^0(x) + \eta_k g(x)$ with the penalty parameter η_k converging to ∞. Examples of penalty functions are $g(x) = \sum_{i\in[m]}[f^i(x)]_+, g(x) = \sum_{i\in[m]}(f^i(x))^2$. (This method is also very closely related to the dual approach, in particular, the Augmented Lagrangian method [Pow69].) Rather, the methods ensure the convergence of the *average* constraint violation over the iterates to zero. Similarly, risk-aware contextual bandits and bandits with knapsack constraints [SDK17; MJY12; JHA15] consider unknown constraint functions with a budget limit. Here, safety refers to ensuring that the total usage of a commodity, e.g., budget for adverts, summed over the sequence of iterates, remains below a threshold. Similar to [YNW17; YN16], the above approaches bound average constraint violation rather than avoiding violation at each iteration. While such a formulation can be well-suited in certain problems such as adverts, it may not be well-suited for safe learning applications discussed above because, in this latter case, constraints need to be satisfied at each step.

Interior-point methods: The Interior Point Method (IPM) is another approach of dealing with constrained optimization, similarly to penalty methods replacing the constrained optimization problem with a sequence of unconstrained subproblems. In contrast to penalty methods, IPM is a feasible optimization approach by definition. That is, we replace the constrained problem (P) by an unconstrained barrier subproblem

$$B_\eta(x) := f^0(x) + \eta g(x). \tag{2.7}$$

The barrier function $g(x)$ has to grow to infinity from the inside of $\mathcal{X}$ when any of the constraint functions $f^i(x)$ converges to 0. Examples of the barrier functions are $g(x) = \sum_{i\in[m]} -\log(-f^i(x)), g(x) = \sum_{i\in[m]} \frac{1}{-f^i(x)}$. By using self-concordance properties of specifically chosen barriers and second-order information, IPM is highly efficient in solving Linear Programming (LP), Quadratic Programming (QP), and Conic optimization problems [AHR12]. However, building the barrier with self-concordance properties is not possible for the unknown constraints. Hinder and Ye [HY19] analyze the trust region method in application to the logarithmic barrier independently of the problem and additionally analyze the gradient descent based approach to solving the log barrier optimization (in the deterministic case).

2.2.2 Zeroth-order stochastic optimization

Known constraint set $\mathcal{X}$ The first-order optimization approaches often can be extended to the zeroth-order information case. Let us start with smooth zeroth-order optimization with constraints. For *convex* problems, Flaxman, Kalai, and McMahan [FKM05] proposes an algorithm achieving a sample complexity of $O(\frac{d^2}{\varepsilon^4})$ using projections.

Since the projections might be computationally expensive, in the projection-free setting, [CZK19] proposes an algorithm achieving a sample complexity of $O(\frac{d^5}{\varepsilon^5})$ for stochastic optimization. Garber and Kretzu [GK20] improve the bound for projection-free methods to $O(\frac{d^4}{\varepsilon^4})$ sample complexity. Instead of projections, both above works require solving Linear Programming (LP) sub-problems at each iteration.

Bubeck, Lee, and Eldan [BLE17] proposes a kernel-based method for adversarial learning achieving $O(d^{9.5}T^{1/2})$ regret, and conjecture that a modified version of their algorithm can achieve $O(\frac{d^3}{\varepsilon^2})$ sample complexity for stochastic black-box convex optimization. At each iteration $t > 0$, this method requires sampling from a specific distribution p_t, that can be done in $\text{poly}(d, \log(T))T$-time. For the smooth and strongly-convex case, Hazan and Luo [HL16] propose a method that achieves $O(\frac{d^3}{\varepsilon^2})$. The general lower bound for the convex black-box stochastic optimization $O\left(\frac{d^2}{\varepsilon^2}\right)$ is proposed by Shamir [Sha13]. Up to our knowledge, there is no proposed lower bound for the safe black-box optimization with unknown constraints.

For *non-convex* optimization, Balasubramanian and Ghadimi [BG18] provide a comprehensive analysis of the performance of several zeroth-order algorithms allowing two-point bandit feedback. However, they also require complete knowledge of the constraints.

There exist also other classical derivative-free optimization methods addressing non-convex optimization based on various heuristics. One example is the Nelder-Mead approach, also known as simplex downhill [NM65]. To handle constraints it uses penalty functions [LLG04], or barrier functions [Pri19]. Another example are various evolutionary algorithms [SP97; KE95; Rec89; HO01]. But all of these approaches are based on heuristics and, thus, do not provide the theoretical convergence rate guarantees; at best, they provide the asymptotic convergence.

Unknown constraint set $\mathcal{X}$ There are much fewer works on safe learning for problems with *unknown* constraints.

Bayesian optimization: The problem of safe learning using Gaussian processes (GP) has been proposed in [Sui+15a]. The SafeOpt algorithm developed in the above work considers minimizing an unknown loss function iteratively while ensuring that the loss of each iterate is above a required threshold. Given actively taken measurements of the loss, the initial estimate of the feasible set is incrementally enlarged through exploration and considering certain regularities of GP kernels. This framework is extended to multiple constraints and experimentally validated on robotic platforms by [BKS16]. Safe GP learning is powerful as it can address general non-convex problems. Nevertheless, due to this generality, current approaches do not scale well with the problem dimension.

A significant line of work covers objectives and constraints with bounded reproducing kernel Hilbert space (RKHS) norm [Sui+15b; BKS16], based on Bayesian Optimization (BO). Also, for *linear* bandits problem Amani, Alizadeh, and Thrampoulidis [AAT19], design a Bayesian algorithm handling the safety constraints. These works build Bayesian models of the constraints and the objective using Gaussian processes [RW05, GP] and crucially require a suitable GP prior. In contrast, in our work, we *do not* use GP models and do not require a prior model for the functions. Additionally, most of these approaches do not scale to high-dimensional problems. Kirschner, Mutny, Hiller, Ischebeck, and Krause [Kir+19] proposes an adaptation to higher dimensions using the line search called LineBO, which demonstrates strong performance in safe and non-safe learning in practical applications. However, they derive the convergence rate only for the unconstrained case, whereas for the constrained case, they only prove safety without convergence. We compare our approach with their method in high dimensions empirically and demonstrate that our approach can solve the problems in cases where LineBO struggles.

Safe Frank-Wolfe method for safe learning (Chapter 3): From the optimization side, in the case of *unknown* constraints, projection-based optimization techniques or Frank-Wolfe-based are not valid without assuming a model of the constraint set. Indeed, such approaches require solving subproblems with respect to the constraint set. Thus, the learner has to know at least an approximate model of it. One can build such a model in the special case of polytopic constraints. In Chapter 3, we propose a safe algorithm called Safe Frank-Wolfe (SFW) for convex learning with smooth objective and *linear* constraints based on the Frank-Wolfe algorithm, that uses the robust optimization technique to address uncertainty. Given a confidence level $1-\delta$ and accuracy ε, we prove that after $\tilde{O}\left(\frac{1}{\varepsilon}\right)$ iterations (one exact objective gradient measurement per iteration) and $\tilde{O}\left(\frac{d^2}{\varepsilon^2}\right)$ zeroth-order constraints measurements in total, the final point is an ε-accurate solution with high probability (Theorem 2). We also conjecture that in case if stochastic first-order measurements are available for the objective, then we require $\tilde{O}\left(\frac{d^2}{\varepsilon^3}\right)$ measurements in total. Similarly, in the case when two-point zeroth-order noisy information is available for the objective, we also require $\tilde{O}(\frac{d^2}{\varepsilon^3})$ objective and constraints noisy value measurements in total. Building on the above approach, [Fer+20] propose an algorithm for both convex and non-convex objective and *linear* constraints. This method considers first-order noisy objective oracle and zeroth-order noisy constraints oracle.

Log barriers SGD for safe learning (Chapter 4): For the more general case of non-linear programming, we propose to use safe optimization approaches based on the interior point method (IPM). Note that primal IPM is a feasible optimization approach by definition. Recall that we cannot construct barriers with self-concordance properties for unknown constraints. Therefore, we focus on logarithmic barriers. Our approach is built on the idea of Hinder and Ye [HY19] who proposes the analysis of the gradient-based approach to solving the log barrier optimization (in the deterministic case). In the current dissertation, we extend the above work to the smooth problems with both first-order and zeroth-order stochastic information. For the zeroth-order setting our method achieves the sample complexity of $\tilde{O}(\frac{d^2}{\varepsilon^7})$ for non-convex problems, $\tilde{O}(\frac{d^2}{\varepsilon^6})$ for convex problems, and $\tilde{O}(\frac{d^2}{\varepsilon^5})$ for strongly-convex problems. We additionally propose the extension to the non-smooth setting using the randomized smoothing technique. We summarize the discussion of the zeroth-order algorithms from the past work as well as the best known lower bounds in Table 2.1.

Price of safety To finalize Table 2.1, in case of polytopic constraints, our method with two-point zeroth-order feedback on the objective and one-point zeroth-order feedback on the constraints requires $O(d^2\varepsilon^{-3})$ measurements compared to the similar zeroth-order conditional gradient method in [BG18] $O(d\varepsilon^{-3})$ (without acceleration or variance reduction) also given two-point zeroth-order queries for the objective and fully *known* constraints. The required number of LP subproblems to be solved in both methods is the same $O(\varepsilon^{-1})$. That is, in this case, we pay only a price of $O(d)$.

As for the non-linear constraints, compared to the state-of-the-art works with tractable algorithms and *known* constraints [BP16], we pay a price of order $\tilde{O}(\varepsilon^{-3})$ in zeroth-order optimization just for the *safety* with respect to unknown constraints both in convex and strongly-convex cases. In the non-convex case, we pay $O(d\varepsilon^{-3})$ both for safety and having one-point feedback compared to Balasubramanian and Ghadimi [BG18] considering 2-point feedback. As for the computational complexity, our log barrier method is projection-free and does not require solving any subproblems compared to the above methods.

Algorithm	**Sample complexity**	**Computational complexity**	**Constraints**	**Convexity**
Bach and Perchet [BP16]	$O\left(\frac{d^2}{\varepsilon^3}\right)$	projections	known	yes
Bach and Perchet [BP16]	$O\left(\frac{d^2}{\mu\varepsilon^2}\right)$	projections	known	μ-str.-convex
Bubeck, Lee, and Eldan [BLE17]	$O\left(\frac{d^3}{\varepsilon^2}\right)$	samplings from p_t distribution	known	yes
Balasubramanian and Ghadimi [BG18]	$O\left(\frac{d}{\varepsilon^4}\right)$	$O\left(\frac{1}{\varepsilon^2}\right)$ LPs	known	no
Garber and Kretzu [GK20]	$O\left(\frac{d^4}{\varepsilon^4}\right)$	$O\left(\frac{1}{\varepsilon^2}\right)$ LPs	known	yes
This work [UKK19]	$O\left(\frac{d^2}{\varepsilon^3}\right)$	$O\left(\frac{1}{\varepsilon}\right)$ LPs	unknown, linear	yes
Fereydounian, Shen, Mokhtari, Karbasi, and Hassani [Fer+20]	$\tilde{O}\left(\frac{d^2}{\varepsilon^4}\right)$	$O\left(\frac{1}{\varepsilon^2}\right)$ LPs	unknown, linear	yes/no
Berkenkamp, Turchetta, Schoellig, and Krause [Ber+17]	$\tilde{O}\left(\frac{\gamma(d)}{\varepsilon^2}\right)$	$O\left(\frac{\gamma(d)}{\varepsilon^2}\right)$ NLPs	unknown	no
This work [Usm+22]	$O\left(\frac{d^2}{\varepsilon^7}\right)$	$O\left(\frac{1}{\varepsilon^3}\right)$ gradient steps	unknown	no
This work [Usm+22]	$\tilde{O}\left(\frac{d^2}{\varepsilon^6}\right)$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ gradient steps	unknown	yes
This work [Usm+22]	$\tilde{O}\left(\frac{d^2}{\mu\varepsilon^5}\right)$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ gradient steps	unknown	μ-str.-convex
Lower bound [Sha13]	$O\left(\frac{d^2}{\varepsilon^2}\right)$	-	known	yes

Table 2.1: Zeroth-order safe smooth optimization algorithms. Here ε is the target accuracy, and d is the dimension of the decision variable. Many of the cited works provide bounds in terms of regret, which can be converted to stochastic optimization accuracy. In SafeOpt, $\gamma(d)$ depends on the kernel, and might be exponential in d. All the above works consider one-point feedback, except for Balasubramanian and Ghadimi [BG18], which considers two-point feedback. In [UKK19] we consider zeroth-order two-point feedback for the objective similarly to [BG18]. In [Fer+20] the authors consider first-order feedback for the objective. In Bubeck, Lee, and Eldan [BLE17], at each iteration the sampling from a a specifically updated distribution p_t can be done in poly$(d, \log(T))T$-time. Under $\tilde{O}(\cdot)$, we hide a multiplicative logarithmic factor.

CHAPTER 3

Safe Convex Learning with Polytopic Constraints

In this chapter, we address the problem of minimizing a convex smooth function $f(x)$ over a compact polyhedral set $\mathcal{X}$ given a stochastic zeroth-order constraint feedback model. This problem arises in safety-critical machine learning applications, such as personalized medicine and robotics. In such cases, one needs to ensure constraints are satisfied while exploring the decision space to find optimum of the loss function. We propose a new variant of the Frank-Wolfe algorithm, which applies to the case of uncertain linear constraints. Using robust optimization, we provide the convergence rate of the algorithm while guaranteeing feasibility of all iterates, with high probability. This chapter is based on our paper Usmanova, Krause, and Kamgarpour [UKK19].

Our contributions We propose an algorithm for safe learning, given a smooth convex objective and a set of unknown linear constraints with noisy oracle information. Given a confidence level $1-\delta$, we ensure feasibility of the iterates with probability at least $1-\delta$ (Theorem 1). Furthermore, given accuracy ε, we prove that after $\tilde{O}\left(\frac{1}{\varepsilon}\right)$ iterations and $\tilde{O}\left(\frac{d^2 \ln 1/\delta}{\varepsilon^2}\right)$ constraints measurements, the final point is an ε-accurate solution with probability $1-\delta$ (Theorem 2). By $\tilde{O}(\cdot)$ we denote $O(\cdot)$ up to a logarithmic multiplicative factor. While in this chapter we mainly focus on the exact first-order oracle for the objective function, we discuss extensions to a stochastic oracle in Section 3.4.

The core idea of our algorithm is to combine a first-order feasible optimization approach with the robust optimization technique. Particularly, our algorithm is based on the Frank-Wolfe (FW) method. In each iteration, it solves an *uncertain* linear program based on estimates of the constraints. Then, it uses this solution to define the step direction. The safety of the iterates is ensured with high probability by refining the confidence set of the unknown parameters iteratively. We emphasize that while we use the theory of robust optimization [BN98; BN99; BN00], our problem formulation is different than that of a classical robust optimization. Specifically, we consider gathering information online about the uncertainty, whereas the robust optimization works assume one-shot knowledge of uncertainties. We numerically evaluate the performance of the proposed algorithm in Section 3.5 and compare its performance with a one-shot robust optimization approach.

3.1 Problem Formulation

The problem of safe learning in its most general form can be defined as a constrained optimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & f^0(x) \\ \text{subject to } & f^i(x) \leq 0 \ \ \forall i \in [m], \end{aligned}$$

where the objective function $f^0 : \mathbb{R}^d \to \mathbb{R}$ and the constraints $f^i : \mathbb{R}^d \to \mathbb{R}$ are unknown, and can only be accessed at feasible points x. The objective is to design an iterative algorithm that chooses the query points to ensure feasibility at each round while progressing towards the optimum. In this chapter we focus on the special case of polytopic constraints in the form $Ax - b \leq 0$. Throughout this chapter we denote the objective function by f instead of f^0. We define all the assumptions below.

Assumptions In this chapter, we consider an instance of the safe learning problem in which the objective f is convex and M-Lipschitz continuous, that is, $|f(x) - f(y)| \leq M\|x - y\| \ \forall x, y \in \mathcal{X}$, where $\mathcal{X}$ is the feasible set. Furthermore, we assume f is L-smooth, that is, f has L-Lipschitz continuous gradients in $\mathcal{X}$, $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathcal{X}$. We assume access to the gradients of the objective function, $\nabla f(x)$, at any feasible query point $x \in \mathcal{X}$. We furthermore assume that constraints are known to be linear, $f_i(x) = [a^i]^T x - b^i$ for $i \in [m]$. Hence, letting $A \in \mathbb{R}^{m \times d}$ denote the matrix with rows defined by $[a^i]^T$, the problem is given by

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \ & f(x) \\ \text{s.t.} \ \ & Ax - b \leq 0. \end{aligned} \tag{3.1}$$

We assume that the feasible set $\mathcal{X} = \{x \in \mathbb{R}^d : Ax - b \leq 0\}$ is a compact polytope with non-empty interior. Denote by D the diameter of the set $\mathcal{X}$, $D = \max_{x,y \in \mathcal{X}} \|x - y\|$. Furthermore, let D_0 be the radius of the smallest ball centered at 0 such that $\mathcal{X} \in \mathcal{B}^d(0, D_0)$, namely, $D_0 = \max_{x \in \mathcal{X}} \|x\|$.

If A and b are known, (P) can be solved efficiently by off-the-shelf first-order convex optimization algorithms. We however, consider the case in which A and b are unknown and can be accessed through an oracle. Specifically, we assume the constraints can be evaluated at any point that lies within a ball of radius ν of the feasible set. These evaluations are corrupted by Gaussian noise. Hence, we have access to $y(x) = Ax - b + \xi$ for any x such that $\mathcal{B}^d(x, \nu) \cap \mathcal{X} \neq \emptyset$, where ξ are sub-Gaussian. If in the problem setting having all the measurements inside the feasible set is critical, we can artificially shrink the set $\mathcal{X}$ by the value ν from the boundaries. This can be achieved by tightening the constraints $[a^i]^T x \leq b^i$ with setting the measurements $\hat{y}^i = y^i - \kappa = [a^i]^T x - b^i + \xi^i - \kappa^i$, with $\kappa^i \geq L_A^i \nu$, where L_A^i is an upper bound on $\|a^i\|$.

The scope of the present chapter is to design an algorithm which, starting from a feasible point $x_0 \in \mathcal{X}$, converges to an optimal solution x_* with a required accuracy ε and a required confidence $1-\delta$ after T steps, that is,

$$\mathbb{P}\{f(x_T) - f(x_*) \leq \varepsilon\} \geq 1-\delta. \tag{3.2}$$

Since the constraint set $\mathcal{X}$ is unknown and revealed through a noisy oracle, we can at the very best ensure to remain inside the feasible set with sufficiently high probability. Hence, we require that the updates of the method are not violating the true constraints with the same required confidence level of $1-\delta$, that is,

$$\mathbb{P}\{Ax_t - b \leq 0,\ 0 \leq t \leq T\} \geq 1-\delta. \tag{3.3}$$

Some words on the choices of the optimization and oracle above are in order. First, the setting of linear constraints can be restrictive for some real-world problems. Nevertheless, understanding the linear setup is often the first step in addressing more challenging formulations. Second, having a noisy first-order or a zeroth-order oracle for the objective function is more realistic for several safe learning problems. Optimization under such oracle models have been deeply explored for the case in which the constraint set is known. Hence, the main novelty and challenge in safe learning is ensuring feasibility of the iterates despite uncertain and incrementally revealed constraint values. We discuss how the proposed algorithm can be generalized to stochastic oracle models for objective in Section 3.4.

3.2 The SFW algorithm

We propose a variant of the Frank-Wolfe algorithm where we explicitly take into account the uncertainty about the feasible set $\mathcal{X}$, referred to as Safe Frank-Wolfe (SFW). The algorithm can be summarized as follows. Starting with a feasible point $x_0 \in \mathcal{X}$, at each iteration $t = 0, \ldots, T$ we generate a number n_t of query points and obtain noisy measurements of the constraint functions at these points. Using linear regression, we obtain an estimate $\hat{\mathcal{X}}_t$ of the feasible set based on the history of obtained measurements. The algorithm then uses $\hat{\mathcal{X}}_t$ to obtain a direction $\hat{s}_t$ by solving the estimated Direction Finding Subproblem (DFS)

$$\hat{s}_t = \arg\min_{s \in \hat{\mathcal{X}}_t} \langle \nabla f(x_t), s \rangle. \tag{3.4}$$

The next iterate is then given by $x_{t+1} = x_t + \gamma_t(\hat{s}_t - x_t)$, according to a chosen step-size γ_t. Below, we further describe each step of the proposed algorithm.

Taking Measurements. During each iteration t of the algorithm, we first make measurements at n_t number of points $x_{(j)}$ within distance ν of x_t in d linearly independent directions. The number n_t needs to satisfy a lower bound as a function of the input data δ, T, to ensure safety. This bound is provided in Theorem 1. Denote by $X_t = [x_{(1)}, \dots, x_{(N_t)}]^T \in \mathbb{R}^{N_t \times d}$ and by $N_t = \sum_{k=0}^{t} n_k$, the total number of available measurements at iteration t. Combining all measurements taken up to iteration t we have the following information about the constraints $y^i = X_t a^i - b^i \mathbf{1} + \xi^i$, $i \in [m]$, where $y^i \in \mathbb{R}^{N_t}$ is the vector of N_t measurements of i-th constraint, $\xi^i = [\xi^i_1, \dots, \xi^i_{N_t}]^T \in \mathbb{R}^{N_t}$ is the vector of errors. The errors ξ^i_t are independent and σ-sub-Gaussian, which means

$$\forall \lambda \in \mathbb{R} \ \forall i \ \mathbb{E}\left[e^{\lambda \xi^i_t}\right] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right).$$

The sub-Gaussian condition implies that $\mathbb{E}\left[\xi^i_t\right] = 0$ and $\operatorname{Var}\left[\xi^i_t\right] \leq \sigma^2$. An example of σ-sub-Gaussian ξ^i are independent zero-mean Gaussian random variables with variance at most σ^2, or independent bounded zero-mean variables lying in an interval of length at most 2σ. We also denote by $\mathbf{1} \in \mathbb{R}^{N_t}$ the vector of 1's. Let us denote by $Y_t = [y^1, \dots, y^m] \in \mathbb{R}^{N_t \times m}$ the matrix of corresponding measurements of the constraints.

Estimating Constraints. Let $\beta^i = \begin{bmatrix} [a^i]^T & b^i \end{bmatrix}^T \in \mathbb{R}^{d+1}$ denote the vector corresponding to the i-th constraint. We refer to β^i as the true parameter. Let $\bar{X}_t = [X_t, -\mathbf{1}] \in \mathbb{R}^{N_t \times (d+1)}$ be the extended version of the matrix X_t. The Least Squares Estimation (LSE) of the constraint parameters at step t is given by

$$\hat{\beta}_t = [\hat{A}_t, \hat{b}_t]^T = [\bar{X}_t^T \bar{X}_t]^{-1} \bar{X}_t^T Y_t. \tag{3.5}$$

The covariance matrix of the $\hat{\beta}^i_t$ is given by $\Sigma_t = \sigma^2 [\bar{X}_t^T \bar{X}_t]^{-1}$. Let $\hat{a}^i_t, \hat{b}^i_t$ denote the estimates of the corresponding rows of $\hat{\beta}^i_t$ and $\hat{\mathcal{X}}_t = \{x \in \mathbb{R}^d : \hat{A}_t x \leq \hat{b}_t\}$ denote the estimated feasible set.

Stopping criteria. Recall that $\hat{s}_t$ is the minimizer of the estimated DFS Eq. (3.4) and let $\hat{g}_t$ be its optimal value

$$\hat{g}_t = \min_{s \in \hat{\mathcal{X}}_t} \langle s, \nabla f(x_t) \rangle. \tag{3.6}$$

Similarly, let s_t denote the minimizer of the DFS under true constraints and g_t the corresponding optimal value

$$s_t = \arg\min_{s \in \mathcal{X}} \langle s, \nabla f(x_t) \rangle, \quad g_t = \min_{s \in \mathcal{X}} \langle s, \nabla f(x_t) \rangle. \tag{3.7}$$

From convexity of f, we have $f(x_t) - f(x_*) \leq g_t$. Thus, as discussed in [Jag13], g_t can be taken as a surrogate duality gap and consequently a stopping criterion for the FW algorithm. In our case, the duality gap cannot be computed exactly because the feasible set $\mathcal{X}$ is unknown. Nevertheless, for the random variable $E_t := |\hat{g}_t - g_t|$ describing an error in the gap estimation we can derive a probabilistic upper bound $\bar{E}_t(\delta)$, such that $\mathbb{P}\{E_t \leq \bar{E}_t(\delta)\} \geq 1 - \delta$ (see Proposition 1 Section 3.4). It follows that if $\hat{g}_t + \bar{E}_t(\delta) \leq \varepsilon$, then with probability greater than $1-\delta$ we have $f(x_t) - f(x_*) \le \varepsilon$. Thus, we use $\hat{g}_t + \bar{E}_t(\delta) \le \varepsilon$ as a stopping criterion.

Putting the above few steps together, we present the Safe Frank-Wolfe (SFW) in Algorithm 1.

Algorithm 1 SFW (Safe Frank-Wolfe)

```
Input: $x_0 \in \mathcal{X}$, bound on iterations $T$, accuracy $\varepsilon$, confidence parameter $\delta$, measurement radius $\nu$;
$t \leftarrow 0$; Choose $n_t(\delta, T)$;
while $t \le T$ do
    Pick $2d$ points around the current point $x_t$
        $x_{(N_{t-1}+i)} = x_t + e_i\nu,$
        $x_{(N_{t-1}+2i)} = x_t - e_i\nu, i \in [d],$
    and take $[n_t/2d]$ measurements at each point;
    Obtain the gradient $\nabla f(x_t)$ and the noisy constraint values $y_{(j)}^i = x_{(j)}^T a^i - b^i + \xi_{(j)}^i$ $\forall j = N_t+1, \dots, N_t+n_t, i \in [m]$;
    Compute the LSE of the constraints $\hat{A}_t$ and $\hat{b}_t$ based on Eq. (3.5);
    Solve the estimated DFS (3.4) to obtain $\hat{s}_t$;
    Estimate the duality gap $\hat{g}_t$ (3.6);
    if $\hat{g}_t \le \varepsilon - E_t(\bar{\delta})$ then
        break and return $x_t$;
    end if
    Set $\gamma_t = \frac{1}{t+2}$;
    $x_{t+1} \leftarrow x_t + \gamma_t(\hat{s}_t - x_t)$;
    $t \leftarrow t + 1$.
end while
```

3.3 Safety

In order to ensure safety of the trajectory as per Inequality (3.3) we ensure that each x_{t+1} generated by the algorithm above remains within the feasible set $\mathcal{X}$ with probability $1-\bar{\delta}$, where $\bar{\delta} = \frac{\delta}{T}$. This is achieved using the analysis framework of robust optimization by [BBC11], [BN98]. The safety of each iterate, combined with a union bound, enables us to prove the safety of the sequence $\{x_t\}_{t=1}^T$ with probability $1 - \delta = 1 - \sum_{t=1}^T \bar{\delta}$.

Since the LSE's of the constraint parameters are given by $\hat{\beta}_t^i = [[\hat{a}_t^i]^T, \hat{b}_t^i]^T \in \mathbb{R}^{d+1}$, the confidence set for the vector of true parameters β^i is given by the following ellipsoid: $\mathcal{E}_t^i(\bar{\delta}) = \left\{ z \in R^{d+1} : (\hat{\beta}_t^i - z)^T \Sigma_t^{-1} (\hat{\beta}_t^i - z) \le \phi^{-1}(\bar{\delta})^2 \right\}$, where,

$$\phi^{-1}(\bar{\delta}) = \max\left\{ \sqrt{128 d \log N_t \log\left(\frac{N_t^2}{\bar{\delta}}\right)}, \frac{8}{3}\log\frac{N_t^2}{\bar{\delta}} \right\}$$

for σ-sub-Gaussian noise ξ^i for $N_t e^{-1/16} \geq \bar{\delta}$, [DHK08] [1]. Thus, we have an ellipsoidal uncertainty set centered at $\hat{\beta}_t^i$, such that $\mathbb{P}\{\beta^i \in \mathcal{E}_t^i(\bar{\delta})\} \geq 1-\bar{\delta}$. We define the confidence set $\mathcal{E}_t(\bar{\delta})$ for all parameters β by $\mathcal{E}_t(\bar{\delta}) = \mathcal{E}_t^1(\bar{\delta}/m) \times \ldots \times \mathcal{E}_t^m(\bar{\delta}/m) \subseteq \mathbb{R}^{(d+1)m}$. The confidence set $\mathcal{E}_t(\bar{\delta})$ determines the uncertainty set for constraint parameters β with probability $1-\bar{\delta}$. Indeed, $1 - \mathbb{P}(\exists i : \beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)) = 1 - \mathbb{P}\{\cup_{i=1}^m \{\beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)\}\} \geq 1 - \sum_{i=1}^m \bar{\delta}/m = 1 - \bar{\delta}$.

We define the safety set $S_t(\bar{\delta}) \subset \mathbb{R}^d$ at iteration t as the set of $x \in \mathbb{R}^d$ satisfying the constraints with any true parameter $\beta = [A, b]$ in the confidence set:

$$S_t(\bar{\delta}) = \{x \in \mathbb{R}^d : Ax \leq b, \ \forall [A, b] \in \mathcal{E}_t(\bar{\delta})\}. \tag{3.8}$$

Given that for each constraint β^i our uncertainty set $\mathcal{E}_t^i(\bar{\delta})$ has an ellipsoidal form, the safety set $S_t(\bar{\delta})$ is equivalent to the intersection of a set of second order cone constraints [BEN09]

$$S_t(\bar{\delta}) = \left\{ x \in \mathbb{R}^d : \forall i \in [m] \ \left[[\hat{a}_t^i]^T x - \hat{b}_t^i \right] + \phi^{-1}(\bar{\delta}/m) \left\| \Sigma_t^{1/2} \begin{bmatrix} x \\ -1 \end{bmatrix} \right\| \leq 0 \right\}. \tag{3.9}$$

Fact 1. *From the definition of the confidence and safety sets it readily follows that*

$$\mathbb{P}\{x \in \mathcal{X} \,|\, x \in S_t(\bar{\delta}), \beta \in \mathcal{E}_t(\bar{\delta})\} = 1.$$

Fact 2. *The condition $x_t \in S_t(\bar{\delta})$ is equivalent to*

$$\phi_{\bar{\delta}} \sqrt{\frac{1}{N_t} + (x_t - \bar{x}_t)^T R_t (x_t - \bar{x}_t)} \leq \min_{i \in [m]} \epsilon_t^i,$$

where $\phi_{\bar{\delta}} = \sigma \phi^{-1}(\bar{\delta}/m)$, $\epsilon_t^i = \hat{b}_t^i - [\hat{a}_t^i]^T x_t$, $\bar{x}_t = \frac{X_t^T \mathbf{1}}{N_t}$, and

$$R_t = \Big[\sum_{j=1}^{N_t} (x_{(j)} - \bar{x}_t)(x_{(j)} - \bar{x}_t)^T \Big]^{-1}.$$

The proof of this fact is provided in Section A.1.

To state the main result on safety of each iteration, we need to introduce some notation. For the polytope $\mathcal{X} \in \mathbb{R}^d$, by an active set B we denote a set of indices of d linearly independent constraints active in a vertex $V \in \mathbb{R}^d$ of $\mathcal{X}$, i.e., $V = V^B = [A^B]^{-1} b^B$. Here, A^B is a corresponding sub-matrix of A and b^B is the corresponding right-hand-side of the constraint. Let $\rho_{\min}(A^B)$ denote the smallest singular value of A^B. Let $Act(\mathcal{X})$ denote the set of all active sets corresponding to vertices of $\mathcal{X}$, i.e., $Act(\mathcal{X}) = \{B : V^B \text{ is a vertex of } \mathcal{X}\}$. Furthermore, define $\rho_{\min}(\mathcal{X}) := \min\{\rho_{\min}(A^B) : B \in Act(\mathcal{X})\}$. Note that $\rho_{\min}(\mathcal{X}) > 0$ since by definition, B is a set of linearly independent active

[1] In the case when the noise is Gaussian and X_t is chosen deterministically, e.g. if all the samples are taken around x_0, $\phi^{-1}(\bar{\delta})$ is taken as the inverse of Chi-squared cumulative distribution function with $d+1$ degrees of freedom [DS14].

constraints. Let $\epsilon_0 = \min_i \{b^i - [a^i]^T x_0\}$, and $L_A = \max_i \|a^i\|$. With the notation in place, we can present the following lemma on the lower bound on the number of measurements to ensure safety of each iterate.

Lemma 1. *If $\beta \in \mathcal{E}_k(\bar{\delta})$ for $k \in [t]$ and $n_t = 4C_n t(\ln t)^2$, with the constant parameter C_n satisfying*

$$C_n \geq C_{\bar{\delta}}^2 \max\left\{\frac{4(\ln\ln T)^2 L_A^2}{[\epsilon_0]^2}, \frac{1}{(D_0+1)^2}\right\}, \tag{3.10}$$

$$\textit{where } C_{\bar{\delta}} = \frac{2\phi_{\bar{\delta}} d(D_0+1)}{\rho_{\min}(\mathcal{X})}\sqrt{\frac{D_0^2+1}{\nu^2}+1}, \tag{3.11}$$

then $x_t \in S_t(\bar{\delta})$. Furthermore, the total number of measurements then satisfies $N_t = C_n t^2(\ln t)^2$.

We provide the full proof in Section A.3.

Proof sketch. Let us give a brief intuition for the proof. First, from Fact 2 we see that in order to have $x_t \in S_t(\bar{\delta})$ we require $\min_i \epsilon_t^i \geq \Omega\left(\frac{1}{\sqrt{N_t}}\right)$, where ϵ_t^i is equal to the distance to the boundary corresponding to the estimated i-th constraint multiplied by $\|\hat{a}_t^i\|$. Second, if these estimates were fixed, then using step sizes $\gamma_t = \frac{1}{t+2}$ we could ensure that the convergence to any boundary i would not be faster than $\epsilon_t^i \geq \prod_{k=0}^t (1-\gamma_k)\epsilon_0^i = \frac{\epsilon_0^i}{t+2}$ (see Figure 3.1). Hence, we require $N_t \geq \Omega\left(\frac{t^2}{\epsilon_0^2}\right)$. However, since ϵ_t^i are random and boundaries are fluctuating, we need N_t to be square logarithmic times larger than the above estimate (as shown in the full proof). ■

Remark. Note that the dependence on $\ln\ln T$ is very mild because this term grows extremely slowly, i.e, $\ln\ln 15 \approx 1$ and $\ln\ln 2000 \approx 2$.

Having established Lemma 1, we can present the safety guarantee of the SFW algorithm.

Theorem 1. *If $n_t = 4C_n t(\ln t)^2$, where the constant parameter C_n is defined in Eq. (3.10) then, the sequence of iterates $\{x_t\}_{t=0}^T$ of SFW is feasible with probability at least $1-\delta$, that is, $\mathbb{P}\{x_t \in \mathcal{X} \text{ for all } 0 \leq t \leq T\} \geq 1-\delta$.*

Proof. Let $\mathcal{F}_t$ denote the event that all the estimated confidence sets $\mathcal{E}_k(\bar{\delta})$ up to step t cover β, i.e., $\mathcal{F}_t = \{\beta \in \cap_{k=0}^t \mathcal{E}_k(\bar{\delta})\}$. Furthermore, let $\mathcal{Q}_t$ denote the event that all the $x_k \in S_k(\bar{\delta})$ up to iteration t, i.e, $\mathcal{Q}_t = \{x_k \in S_k(\bar{\delta}) \text{ for all } 0 \leq k \leq t\}$. By Lemma 1 if $\mathcal{F}_t$ holds, then $x_t \in S_t(\bar{\delta})$. Hence, it is easy to see that $\mathcal{F}_t$ implies $\mathcal{Q}_t$, i.e., $\mathbb{P}\{\mathcal{Q}_t|\mathcal{F}_t\} = 1$. Thus, using Fact 1 we derive

$$\mathbb{P}\left\{x_t \in \mathcal{X} \text{ for all } 0 \leq t \leq T\right\} = \mathbb{P}\{\mathcal{Q}_T, \mathcal{F}_T\} = \mathbb{P}\{\mathcal{F}_T\}.$$

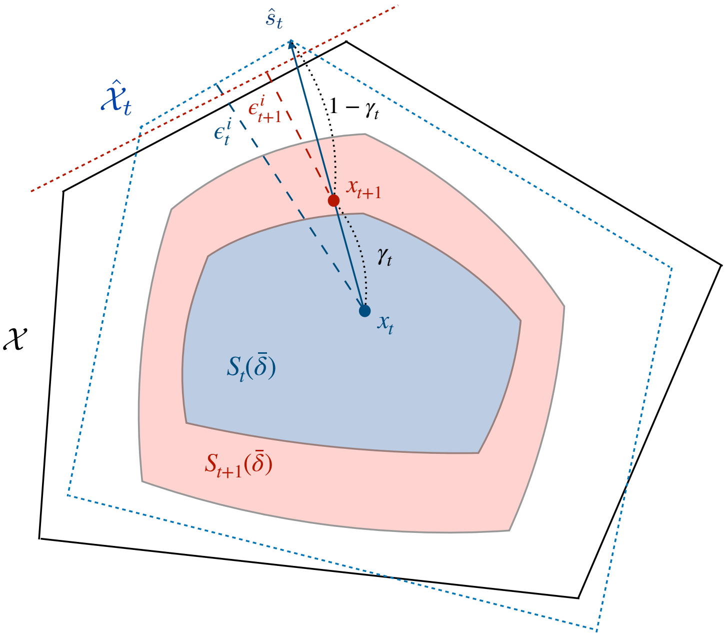


Figure 3.1: Illustration of one iteration of SFW $x_{t+1} = x_t + \gamma_t(\hat{s}_t - x_t)$. Bold lines denote the polytope $\mathcal{X}$ and dashed lines denote its estimate $\hat{\mathcal{X}}_t$.

Using Boole's inequality we can bound the probability of $\mathcal{F}_T$ as follows

$$\mathbb{P}\{\mathcal{F}_T\} = 1 - \mathbb{P}\{\cup_{t=0}^{T} \cup_{i=1}^{m} \{\beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)\}\} \geq 1 - \sum_{t=0}^{T}\sum_{i=1}^{m} \bar{\delta}/m \geq 1 - T\bar{\delta}.$$

This concludes the proof. ■

3.4 Convergence

First, we show that the proposed algorithm achieves the optimal convergence rate for the Frank-Wolfe algorithm (see [Lan13] , with sufficiently high probability. Second, we discuss extensions to stochastic first-order and zeroth-order oracles of the objective function based on the results of [HL16].

Convergence rate Let us define the curvature constant C_f of the function $f(x)$ with respect to the compact domain $\mathcal{X}$ by

$$C_f = \sup_{\substack{x,s\in\mathcal{X},\gamma\in[0,1],\\ y=x+\gamma(s-x)}} \frac{1}{\gamma^2}(f(y) - f(x) - \langle y - x, \nabla f(x)\rangle).$$

It can be verified that $C_f \leq LD^2$, where L is the Lipschitz constant of the gradient $\nabla f(x)$ and D is the diameter of the set $\mathcal{X}$ (see Section 3.1). Our main result is as follows.

Theorem 2. *If $n_t = 4C_n(t+2)(\ln(t+2))^2$ and C_n is chosen according to the bound in Equation Eq. (3.10), then:*
a) after T steps of the SFW algorithm, the final point x_T satisfies

$$\mathbb{P}\left\{ f(x_T) - f(x_*) \leq \frac{f(x_0) - f(x_*)}{T+2} + \frac{\ln{(T+2)}\frac{C_f}{2} + \ln\ln(T+2)\frac{C'}{2}}{T+2} \right\} \geq 1 - \delta,$$

where $C' = \frac{MC_{\bar{\delta}}}{\sqrt{C_n}}$, and $C_{\bar{\delta}}$ is defined in Eq. (3.11).
b) all the iterates $\{x_t\}_{t=1}^T$ are feasible with probability $1-\delta$, as required in Eq. (3.3).

Corollary 1. *The SFW algorithm achieves an ε-accurate solution with probability greater than $1-\delta$ after making $\tilde{O}(\frac{1}{\varepsilon})$ linear optimization oracle calls and $\tilde{O}\left(\frac{d^2 \ln \frac{1}{\delta}}{\varepsilon^2}\right)$ zeroth-order inexact constraint oracle calls.*

Below, we provide the proof sketch for Theorem 2. The full proofs of Theorem 2 and Corollary 1 are provided in Section A.4 and Section A.5.

Proof sketch. Our proof is based on the extensive study of FW convergence provided by [Jag13], [FG16]. Recall that E_t is the accuracy with which an approximated DFS at iteration t is solved. Similarly to ([FG16], Theorem 5.1) we can show that for $\gamma_t = O\left(\frac{1}{t}\right)$, we have

$$f(x_t) - f(x_*) \leq O\left(\frac{\epsilon_0 + C_f \ln t + \sum_{t=1}^T E_t}{t}\right), \tag{3.12}$$

where $\epsilon_0 = f(x_0) - f(x_*)$. Hence, to prove the convergence rate of the SFW algorithm we need to show that the error in the DFS solution decreases with the rate $O\left(\frac{1}{t}\right)$. This fact is shown in Proposition 1 below.

Proposition 1. *If $\beta \in \mathcal{E}_t(\bar{\delta})$ and $N_t \geq \frac{C_{\bar{\delta}}^2}{(D_0+1)^2}$, then $E_t \leq \frac{MC_{\bar{\delta}}}{\sqrt{N_t}}$. Since $\mathbb{P}\{\beta \in \mathcal{E}_t(\bar{\delta})\} \geq 1-\bar{\delta}$, we obtain $\mathbb{P}\left\{E_t \leq \frac{MC_{\bar{\delta}}}{\sqrt{N_t}}\right\} \geq 1 - \bar{\delta}$.*

We provide the proof in Section A.2. From the result above it directly follows that $E_t = O\left(\frac{1}{t \ln t}\right)$, hence $\sum_{k=0}^t E_k = O\left(\ln \ln t\right)$. Using this result, and the classical FW proof technique ([FG16]) we can conclude the result of Theorem 2. This concludes the proof sketch. ■

We can see that under our choice of the number of measurements n_t at each iteration, the total number of measurements is $O\left(d^2 t^2 \ln t^2\right)$. It follows that the required number of measurements at each step grows almost linearly with the iteration number and quadratically with the dimension d. Note however that the number of iterations is independent of the dimension d. In contrast, the safe learning approach in [Sui+15a] is based on gridding the decision space and hence, the dependence in d is exponential. Hence, compared to previous safe learning approaches [Sui+15a; BKS16], our method scales better with dimension. Naturally, this scalability is due to the assumption of convexity of the cost function and the linearity of the constraints.

Finally, let us clarify some computational complexity aspects. After adding each new data point to X, the matrix inversion $(X^TX)^{-1}$ in Step 6 can be performed using one-rank updates (e.g., using formula $(A+vv^T)^{-1} = A^{-1} - \frac{1}{(1+v^TA^{-1}v)}(A^{-1}vv^TA^{-1})$). The cost of each such operation is $O(d^2)$. This operation is to be made $N_t = O(d^2t^2(\ln t)^2)$ times. The total computational complexity is thus $O(d^2N_t) = O(d^4t^2(\ln t)^2)$ from Step 6 and additionally t LP oracle calls in Step 7.

Extension to stochastic oracle for the objective function. Note that the SFW algorithm requires $t = \tilde{O}\left(\frac{1}{\varepsilon}\right)$ iterations and $N_t = \tilde{O}\left(\frac{d^2\ln\frac{1}{\delta}}{\varepsilon^2}\right)$ measurements of constraints to obtain a required accuracy of ε. General Stochastic Frank-Wolfe algorithm with stochastic objective but known linear constraints require $t = O(\frac{1}{\varepsilon})$ iterations and, in contrast, $t = O(\frac{1}{\varepsilon^3})$ stochastic gradient measurements of the objective ([HL16], Table 2).[2] This difference in the number of measurements is due to the fact that in the absence of linearity of the objective function, the gradients of the objective function are changing in each iteration. Thus, $O(\frac{1}{\varepsilon^2})$ measurements at each iteration are needed to guarantee correct variance reduction rate of the Frank-Wolfe method (see Eq. (3.12)). (Note that we do not consider the variance reduction or acceleration techniques for simplicity, however, using them we could reduce the number of measurements above.) From the above observation, we can extend the SFW analysis to the case in which we have access to a stochastic first-order oracle of the objective function. In this case, a total of $O(\frac{1}{\varepsilon^3})$ calls to the objective function oracle, and $O(\frac{d^2}{\varepsilon^2})$ calls to the constraints oracle are sufficient to obtain the desired rate of decrease of E_t in Proposition 1 and hence, the convergence rate in Theorem 2. That is, we require $\tilde{O}\left(\frac{d^2}{\varepsilon^2}\right)$ constraints measurements and $\tilde{O}\left(\frac{1}{\varepsilon^3}\right)$ objective gradient measurements for our method, that we can roughly bound by $\tilde{O}\left(\frac{d^2}{\varepsilon^3}\right)$.

For the case of two-point zeroth-order oracle, $O(\max\{\frac{d}{\varepsilon^2}, d^2\})$ calls per iteration are needed to estimate the gradient of the objective with accuracy ε without using any variance reduction or acceleration techniques. [3] Therefore, we need $\tilde{O}\left(\frac{d^2}{\varepsilon^2}\right)$ constraints measurements and $\tilde{O}(\frac{d}{\varepsilon^3})$ objective value measurements, that we can roughly bound by $O(\frac{d^2}{\varepsilon^3})$ value measurements in total. Note that stochastic FW approach without acceleration with known constraints and two-point zeroth-order feedback on the objective requires $O(\frac{d}{\varepsilon^3})$ as shown in [BG18]. The noisy objective gradient does not influence the safety of the proposed algorithm. Thus, the safety results in Theorem 3 extend to the case with stochastic first-order or zeroth-order oracle of the objective.

[2] The projection-free scheme called STORC in [HL16] can achieve better rates of $O\left(1/\varepsilon^{1.5}\right)$ stochastic gradient oracle calls using the variance reduction technique, but we do not consider now such extensions for our comparison of the direct safe and unsafe FW approach. Alternatively, we could also use the variance reduction to improve the performance given the stochastic gradients.

[3] This can be shown using Lemma 10, where the last term in the variance expression for the estimated gradient simply equals to 0 in the case of two-point feedback.

3.5 Simulations

We evaluate the performance of the proposed approach experimentally. In the first experiment, we consider the convergence rate of the algorithm as a function of the dimension. In the second experiment, we compare the SFW algorithm with a robust optimization based approach, which first learns the uncertain constraints and then finds the optimum with respect to the estimated constraints. We consider the convex smooth optimization problem:

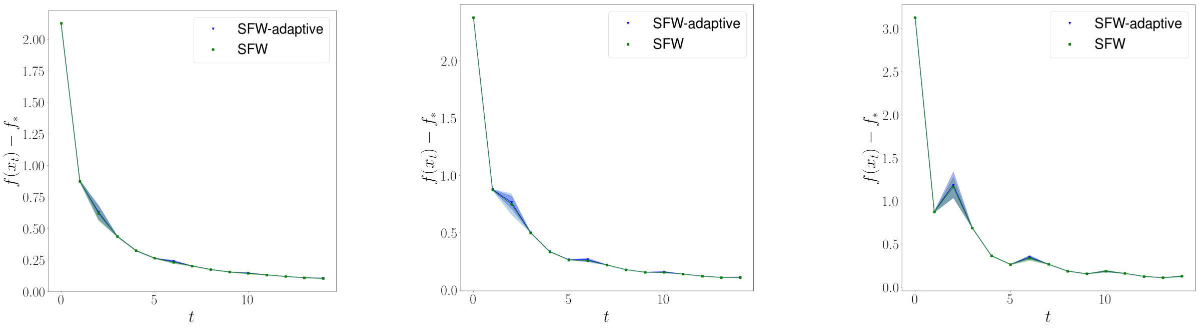
$$\min_{x \in \mathcal{X}} \frac{1}{2} \|x - x'\|_2^2 ,$$

where $\mathcal{X} = \left\{ x \in \mathbb{R}^d | -1 \leq x^i \leq 1, i \in [d] \right\}$ and $x' = [2, 0.5, \ldots, 0.5] \in \mathbb{R}^d$ for varying dimension d. Then, the solution x_* is a point on the boundary of the true constraint set above. We set the variance of the noise to $\sigma = 0.01$ and use a constant exploration radius $\nu = 0.01$. Furthermore, we set the confidence parameter $\delta = 0.1$ and the total number of iterations to $T = 15$. The code corresponding to this experiment can be found under the following link: https://github.com/Ilnura/Thesis_applications.

Empirical constraint violation and convergence rate. The first experiment evaluates the empirical convergence rate and constraint violation as a function of the dimension d. First, we evaluate the convergence rate assuming we can obtain the required lower bound on C_n as per Theorem 2. We run the algorithm for dimensions $d = 2, 10, 20$. In particular, the parameters of the problem required for obtaining the lower bound are derived based on knowledge of the constraints and the objective function, as well as the input parameters δ, T as follows: $\epsilon_0 = 1, \bar{\delta} = 0.0067, \phi_{\bar{\delta}} = 3.43$, $\|b\| = 2, \|a\| = 1$. It follows that a value of $C_n = d^2 \cdot 24$ achieves the required number of measurements. The SFW proposed in Algorithm 1 is then run with the above choices of parameters $\bar{\delta}$ and n_t. For each dimension, we run the SFW algorithm 20 times, keeping all the parameters and the initial conditions the same. The difference in each experiment is due to the stochastic noise in the measurements. The average and standard deviation of the function values $f(x_t) - f(x_*)$ scaled by the initial condition error are shown in Figure 3.2. It can be seen that the dimension does not influence the convergence rate, rather, it influences only the number of measurements.

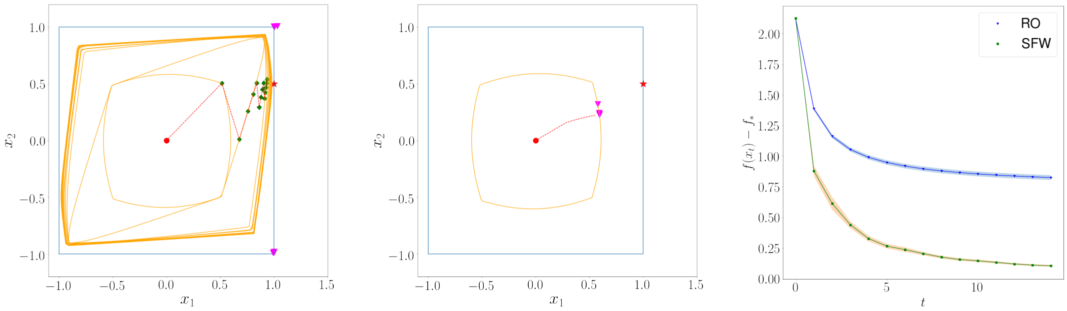
We also run the experiment assuming we cannot compute the lower bound C_n precisely due to lack of problem data. In this case, at each iteration we first take $2dt$ measurements and further continuously take new measurements around x_t until the safety set $S_{t+1}(\bar{\delta})$ grows sufficiently to ensure x_{t+1} becomes safe (see Fact 2). This safety indeed verifies the feasibility of iterates with high probability based on Fact 1. Let us refer to this as the adaptive variant of SFW. This adaptive approach is not only more practical due to lack of dependence on problem data, but also requires far fewer measurements in total since the bound on n_t from the Theorem 1 is quite conservative. This is due to the fact that our theoretical bounds were derived using the worst-case estimate of

$\|\Sigma_t^{1/2}\|$, when the measurements are taken always around the same point. However, $\|\Sigma_t^{1/2}\|$ can reduce much faster in practice. The convergence will also hold since the bound on $E_t \leq \frac{C_{\bar{\delta}} M}{\sqrt{N_t}} \leq \frac{C_{\bar{\delta}} M}{t+2} = O\left(\frac{1}{t}\right)$ required for the convergence rate is still satisfied. In Figure 3.2 caption we reported the required total number of measurements up to step T of the adaptive variant by N_T^a. You can see that it has significantly reduced compared to the non-adaptive variant.



(a)$d = 2, N_T = 8396, N_T^a = 519$ (b) $d = 4, N_T = 16792, N_T^a = 1135$ (c) $d = 10, N_T = 41980, N_T^a = 4275$

Figure 3.2: Convergence rate of SFW method for the dimensions $d = 2, 4, 10$ with $T = 15$.



(a) SFW (b) RO (c) Convergence rates of SFW and RO

Figure 3.3: Trajectories and accuracy of the objective value by each iteration of optimization. The left pair of plots shows the convergence of one realization of SFW method and of the robust optimization approach (without safety set updates) with $\sigma = 0.1$, $T = 15$, $N_T = 5500$. The orange lines denote the boundaries of the safety sets $S_t(\bar{\delta})$. Red circle denotes the starting point and star denotes the solution of the original problem. Magenta triangles denote the estimated DFS solutions.

Comparison with an alternative robust optimization approach. We compare the proposed SFW method with an alternative approach in which we first make enough measurements in the a priori safe region to estimate the safety set (see definition in Eq. (3.8)) with sufficiently high probability. Next, we run a first-order method, such as FW with respect to the nonlinear set $S(\bar{\delta})$. Let us call this approach RO for robust optimization. To compare these two methods we set an a priori number of measurements for the alternative RO method equal to the total number of measurements N_t, of SFW algorithm corresponding to $\delta = 0.1$ and $T = 15$. After estimation of the safety set, we make $T = 15$ iterations of the FW method with the constraint set $S_T(\bar{\delta})$. Thus, the total number of measurement and the total number of optimization steps of the two methods are equal. During the run of the SFW, as per discussion in the above example, we reduce the number of measurements required to ensure safety at each iteration t, online. Figure 3.3(a),(b) shows the optimization trajectories of each method. The green round points along the trajectory correspond to the points where the constraints were measured. We also show the comparison of their convergence rates in Figure 3.3(c). As we can see, SFW algorithm performs better both in terms of estimates of the constraints and convergence rate. This difference in performance can be explained based on two observations. First, SFW moves measurements along the trajectory $\{x_t\}$, and this can lead to smaller variance of the estimates $\Sigma_t = \sigma^2 \left(\bar{X}_t^T \bar{X}_t\right)^{-1}$. Hence, the measurements are more informative and the safety set $S_T(\bar{\delta})$ is larger. Second, SFW algorithm is proven to converge to an ε-optimal solution corresponding to the true constraints. The RO approach however, can at the very best converge to an optimum with respect to a safety set estimated in advance. From the computational perspective, at each iteration the proposed SFW method requires an LP oracle, whereas the alternative RO approach requires solving a second-order cone program. Hence, the SFW is more tractable.

3.6 Conclusion

In this chapter, we proposed a safe learning approach for convex costs and uncertain linear constraints. This method uses information along the optimization trajectory to decrease the objective value and to explore an unknown feasible set. Meanwhile, it ensures feasibility for each iteration with high probability. We provided an analysis of the convergence rate of our algorithm, as well as of feasibility guarantees for its iterations. One open question is how to provide performance guarantees in terms of regret.

In the current work, we do not try to take into account experiment design for better exploration. However, it would be interesting if future to extend the current setting to the online optimization framework and the setting with fully bandit information with regret minimization and to see how constraints exploration - objective exploitation trade-off would play. In the next chapter, our next step is to obtain safe learning results subject to nonlinear constraints.

CHAPTER 4

Safe Non-linear Optimization with Logarithmic Barriers

For safety-critical black-box optimization tasks, observations of the constraints and the objective are often noisy and available only for the feasible points. We propose an approach based on log barriers to find a local solution of a non-convex non-smooth black-box optimization problem $\min f^0(x)$ subject to $f^i(x) \leq 0,\ i = 1, \dots, m$, guaranteeing constraint satisfaction while learning an optimal solution with high probability. Our proposed algorithm exploits noisy observations to iteratively improve on an initial safe point until convergence. We derive the convergence rate and prove safety of our algorithm. We demonstrate its performance in an application to an iterative control design problem.

We introduce a general approach for seeking a stationary point in high dimensional non-linear stochastic optimization problems in which maintaining *safety* during the learning is crucial. Our approach called LB-SGD is based on applying stochastic gradient descent (SGD) with a carefully chosen adaptive step size to a logarithmic barrier approximation of the original problem. We provide a complete convergence analysis of non-convex, convex, and strongly-convex smooth constrained problems, with first-order and zeroth-order feedback. Our approach has efficient updates, scales better with dimensionality compared to existing safe Bayesian optimization approaches.

We empirically compare the sample complexity and the computational cost of our method with other existing safe learning approaches on synthetic benchmarks. As a key case study, in the next Chapter 5 we demonstrate the effectiveness of our approach on minimizing constraint violation in policy search tasks in safe reinforcement learning (RL).

This chapter is based on our paper Usmanova, As, Kamgarpour, and Krause [Usm+22] and uses some parts of Usmanova, Krause, and Kamgarpour [UKK20].

Our contributions We summarize our contributions below:

- We propose *Log Barriers SGD (LB-SGD)*, an algorithm that addresses the safe learning task by minimizing the log barrier approximation of the problem. This minimization is done by using Stochastic Gradient Descent (SGD) with a carefully chosen adaptive step size.

- This approach is unified for safe learning given first-order or zeroth-order stochastic oracle. We prove that our approach generates feasible iterations with high probability. Each iteration of the proposed method is computationally cheap and does not require solving any subproblems such as those required for Frank-Wolfe (LP subproblems) or BO-based algorithms (NLP subproblems).

- We derive the convergence rate of our algorithm for the stochastic non-convex, convex, and strongly-convex problems. We prove the convergence despite the non-smoothness of the log barrier and the increasingly high variance of the log barrier gradient estimator. Furthermore, we propose the extension of our algorithm allowing to address non-smooth problems using the smoothing by randomization technique.

- We demonstrate LB-SGD's performance compared to other safe BO optimization algorithms on a series of experiments with various scales.

4.1 Problem statement

We consider a general constrained optimization problem:

$$\begin{aligned} &\min f^0(x) \\ &\text{s.t } f^i(x) \leq 0, i \in [m], \end{aligned} \tag{P}$$

where the objective function $f^0 : \mathbb{R}^d \to \mathbb{R}$ and the constraints $f^i : \mathbb{R}^d \to \mathbb{R}$ are *unknown*, possibly non-convex functions. We denote by $\mathcal{X}$ the feasible set $\mathcal{X} := \{x \in \mathbb{R}^d : f^i(x) \leq 0, i \in [m]\}$.

Our goal is to solve the *safe learning* problem. That is, we need to find the solution of the constraint problem (P) while keeping all the iterates x_t of the optimization procedure feasible $x_t \in \mathcal{X}$ with high probability during the learning process. Throughout this chapter we make the following assumptions:

Assumption 1. *Set $\mathcal{X}$ has bounded diameter, that is $\exists D > 0$ such that for any $x, y \in \mathcal{X}$ we have $\|x - y\| \leq D$.*

Assumption 2. *The objective and the constraint functions $f^i(x)$ for $i \in \{0, \dots, m\}$ are M_i-smooth and L^i-Lipschitz continuous on $\mathcal{X}$ with constants $L_i, M_i > 0$. We denote by $L := \max_{i=[m]}\{L_i\}$.*

The above two assumptions are standard in the optimization literature.

Assumption 3. *There exists a starting point $x_0 \in \mathcal{X}$ at which $\max_{i \in [m]} f^i(x_0) \leq -\beta$, for $\beta > 0$.*

The third assumption ensures that we have a safe starting point, away from the boundary. In the absence of such an assumption, even the first iterate might be unsafe.

Assumption 4. *There exists $\rho \in (0, \frac{\beta}{2}]$ such that for any point $x \in \mathcal{X}$ there exists a direction $s_x \in \mathbb{R}^d : \|s_x\| = 1$, such that $\langle s_x, \nabla f^i(x)\rangle > l$ with $l > 0$, for all $i \in \mathcal{I}_\rho(x)$ which are ρ-approximately active at x, i.e., $\mathcal{I}_\rho(x) := \{i \in [m] | - f^i(x) \leq \rho\}$.*

The last assumption is the extended Mangasarian-Fromovitz Constraint Qualification (MFCQ). The classic MFCQ [MF67] is the regularity assumption on the constraints, guaranteeing that they have a uniform descent direction for all constraints at a local optimum. The classic MFCQ is satisfied if there exists $s \in \mathbb{R}^d$ such that $\langle s, \nabla f^i(x^*)\rangle < 0$ for all $i \in \mathcal{I}(x^*)$, where x^* is a local minimizer of constrained problem (P), and $\mathcal{I}(x^*) := \{i \in [m] : f^i(x^*) = 0\}$ denotes the set of active constraints at x^*.

Our extended MFCQ guarantees this regularity condition at all points ρ-close to the boundary. This assumption holds for example for convex problems with the constraint set having a non-empty interior, that we show in Section 4.3.3.

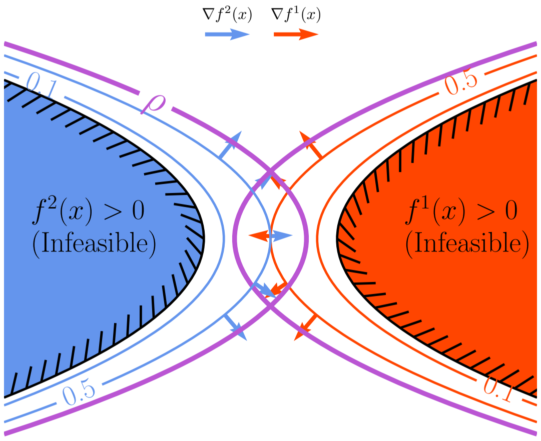


Figure 4.1: Illustration of extended MFCQ.

On Figure 4.1, for the point in the middle, both constraints are ρ-almost active. Note that at this point, no descent direction exists for both constraints since their gradients are pointing to the opposite directions. That is, this set does not satisfy the extended MFCQ with the given $\rho > 0$.

Oracle Typically in the applications we consider, the information available to the learner is noisy. For example, one can only observe the perturbed gradients and values of $f^i, \forall i = 0, \ldots, m$ at the requested points x_t. Therefore, formally we consider access to the first-order stochastic oracle for every $f^i(x)$, providing the pair of value and gradient stochastic estimates:

$$\mathcal{O}(f^i, x, \xi) = (F^i(x, \xi), G^i(x, \xi)). \tag{4.1}$$

Note that the variances of $F^i(x,\xi)$ and $G^i(x,\xi)$ are fixed and given by the nature of the problem. However, we can decrease these variances by taking several measurements per iteration and replacing $(F^i(x,\xi), G^i(x,\xi))$ with

$$F^i_n(x,\xi) := \frac{\sum_{j=1}^n F^i(x,\xi_j)}{n} \text{ and } G^i_n(x,\xi) := \frac{\sum_{j=1}^n G^i(x,\xi_j)}{n}. \tag{4.2}$$

In the above, we abuse the notation and replace the dependence $F^i_n(x,\xi_1,\dots,\xi_n)$ by $F^i_n(x,\xi)$ for simplicity. Then, their variances become respectively such that

$$\mathbb{E}[\|F^i_n(x,\xi) - \mathbb{E}F^i_n(x,\xi)\|^2] \leq \sigma_i^2(n) := \frac{\sigma_i^2}{n}, \tag{4.3}$$

$$\mathbb{E}[\|G^i_n(x,\xi) - \mathbb{E}G^i_n(x,\xi)\|^2] \leq \hat{\sigma}_i^2(n) := \frac{\hat{\sigma}_i^2}{n}. \tag{4.4}$$

Our goal ism given the provided first-order stochastic information, to find an approximate solution of problem (P) while not making value and gradient queries outside the feasibility set $\mathcal{X}$ with high probability. To do so, we introduce the log barrier optimization approach.

4.2 General approach

4.2.1 Safe learning with log barriers

The main idea of the approach is to replace the original constrained problem (P) by it's unconstrained *log barrier* surrogate $\min_{x\in\mathbb{R}^d} B_\eta(x)$ with

$$B_\eta(x) = f^0(x) + \eta\sum_{i=1}^m -\log(-f^i(x)), \tag{4.5}$$

$$\nabla B_\eta(x) = \nabla f^0(x) + \eta\sum_{i=1}^m \frac{\nabla f^i(x)}{-f^i(x)}. \tag{4.6}$$

This approximation $B_\eta(x)$ grows to infinity as the argument converges to the boundary of the set $\mathcal{X}$, and is defined only in the interior of the set $Int(\mathcal{X})$. Therefore, under Assumptions 1 to 4, a major advantage of this method for the problems we consider, is that by carefully choosing the optimization step-size, *the feasibility of all iterates is maintained automatically.* We run Stochastic Gradient Descent (SGD) with the specifically chosen step size applied to the log barrier surrogate $\min_{x\in\mathbb{R}^d} B_\eta(x)$.

The main intuition is that the descent direction of the log barrier pushes the iterates away from the boundary, at the same time converging to an approximate KKT point for the non-convex case, and to an approximate solution for the convex case. To measure the approximation in the non-convex case, we define the *ε-approximate KKT point* (**ε-KKT**).

Specifically, for $\varepsilon > 0$ and a pair (x, λ), such point satisfies the following conditions:

$$\lambda_i, -f^i(x) \geq 0, \ \forall i \in [m] \tag{ε-KKT.1}$$

$$\lambda_i(-f^i(x)) \leq \varepsilon, \ \forall i \in [m] \tag{ε-KKT.2}$$

$$\|\nabla_x \mathcal{L}(x, \lambda)\| \leq \varepsilon. \tag{ε-KKT.3}$$

Whereby, λ is the vector of dual variables and $\mathcal{L}(x, \lambda) := f^0(x) + \sum_{i=1}^m \lambda_i f^i(x)$ is the Lagrangian function of (P). Later, we show the that the SGD on the η-log barrier surrogate converges to an ε-approximate KKT point with $\varepsilon = \eta$. We show it by demonstrating that the small barrier gradient norm $\|\nabla B_\eta(\hat{x})\| \leq \eta$ corresponds to the gradient of the Lagrangian $\|\nabla_x \mathcal{L}(x, \lambda)\|$ with specifically chosen vector of dual variables $\lambda \in \mathbb{R}^m$ [HY19; UKK20]. In the convex case, the approximate optimality in the value $B_\eta(\hat{x}) - B_\eta(x^*_\eta) \leq \eta$ itself implies that $\hat{x}$ is an ε-approximate solution of the original problem: $f^0(\hat{x}) - \min_{x \in \mathcal{X}} f^0(x) \leq \varepsilon$ with $\varepsilon > 0$ linearly dependent on η up to a logarithmic factor.

4.2.2 Main results

We propose to apply SGD with an adaptive step-size to minimize the unconstrained log barrier objective B_η. We name our approach LB-SGD. We show that LB-SGD (with confidence parameter $\delta = \hat{\delta}/Tm$) achieves the following convergence results for the target probability $1 - \hat{\delta}$:

1. For the non-convex case, after at most $T = O(\frac{1}{\varepsilon^3})$ iterations, and with $\sigma_i(n) = O(\varepsilon^2)$ and $\hat{\sigma}_i(n) = O(\varepsilon)$, LB-SGD outputs $x_{\hat{t}}$ which is an ε-KKT point with probability $1 - \hat{\delta}$. For the constant $\hat{\sigma}_i, \sigma_i > 0$, we require $N = Tn = O(\frac{1}{\varepsilon^7})$ oracle queries $\mathcal{O}(f^i, x, \xi)$ for all $i \in \{0, \dots, m\}$. (Theorem 4)

2. For the convex case, after at most $T = \tilde{O}(\frac{\|x_0 - x^*\|^2}{\varepsilon^2})$ iterations of LB-SGD, and with $\sigma_i(n) = \tilde{O}(\varepsilon^2)$ and $\hat{\sigma}_i(n) = \tilde{O}(\varepsilon)$, we obtain output $\bar{x}_T$ such that with probability $1-\hat{\delta}$: $f^0(\bar{x}_T) - \min_{x \in \mathcal{X}} f^0(x) \leq \varepsilon$. For the constant $\hat{\sigma}_i, \sigma_i > 0$, we require $N = Tn = \tilde{O}(\frac{1}{\eta^6})$ calls of the oracle $\mathcal{O}(f^i, x, \xi)$ for all $i \in \{0, \dots, m\}$. (Theorem 5)

3. For μ-strongly-convex case, after at most $T = \tilde{O}(\frac{1}{\mu\varepsilon} \log \frac{1}{\varepsilon})$ iterations of LB-SGD with decreasing η, and with $\sigma_i(n) = \tilde{O}(\eta^2)$ and $\hat{\sigma}_i(n) = \tilde{O}(\eta)$, for the output $\hat{x}_K$ we have with probability $1 - \hat{\delta}$: $f^0(\hat{x}_K) - \min_{x \in \mathcal{X}} f^0(x) \leq \varepsilon$. For the constant $\hat{\sigma}_i, \sigma_i > 0$, we require $N = \tilde{O}\left(\frac{1}{\varepsilon^5}\right)$ calls of the oracle $\mathcal{O}(f^i, x, \xi)$ for all $i \in \{0, \dots, m\}$. (Theorem 6)

4. For the zeroth-order information case, estimating the function gradients using finite difference, we obtain the following bounds on the number of measurements (Corollary 2):

   - $N = O(\frac{d^2}{\varepsilon^7})$ to get an ε-approximate KKT point in the non-convex case;
   - $N = \tilde{O}(\frac{d^2}{\varepsilon^6})$ to get an ε-approximate minimizer in the convex case;
   - $N = \tilde{O}(\frac{d^2}{\varepsilon^5})$ to get an ε-approximate minimizer in the strongly-convex case;

5. We extend this result for the *non-smooth case* with zeroth-order information, we obtain the following bounds on the number of measurements (Corollary 3) to find an ε-approximate solution of the smoothed approximation of the problem (P):
   - $N = O(\frac{d^{2.5}}{\varepsilon^7})$ to get an ε-approximate KKT point in the non-convex case;
   - $N = \tilde{O}(\frac{d^{2.5}}{\varepsilon^6})$ to get an ε-approximate minimizer in the convex case;
   - $N = \tilde{O}(\frac{d^{2.5}}{\varepsilon^5})$ to get an ε-approximate minimizer in the strongly-convex case;
6. In all of the above cases the safety is guaranteed with probability $1 - \delta$ for all the measurements. (Theorem 3, Corollary 2, Corollary 3)

In the above, $\tilde{O}(\cdot)$ denotes $O(\cdot)$ dependence up to a multiplicative logarithmic factor. Note that for zeroth-order information case we only pay the price of a multiplicative factor d^2.

4.2.3 Our approach

To minimize the log barrier function, we employ SGD using the stochastic first-order oracle providing $(F^i(x,\xi), G^i(x,\xi))$ with an adaptive step size, and derive convergence rate of our methods dependent on the noise level of this oracle. At iteration t we make the step in the form:

$$x_{t+1} \leftarrow x_t - \gamma_t g_t, \tag{4.7}$$

where γ_t is a safe adaptive step size, g_t being the log barrier gradient estimator. In Section 4.2.3, we show how to build the estimator g_t of the log barrier gradient. Following that, in Section 4.2.3, we explain how to choose γ_t.

As mentioned before, the log barrier function is not a smooth function due to the fact that close to the boundaries of $\mathcal{X}$ it converges to infinity. To address non-smooth stochastic problems, optimization schemes in the literature typically require bounded sub-gradients. For the log barrier function even this condition does not hold in general. Hence, we cannot expect the classical analysis with the classical predefined step size hold for the SGD applied for the log barrier problem. Contrary to that, by making the step size *adaptive* we can guarantee *local-smoothness* of the log barrier. Intuitively, this is done by restricting the growth of the constraints. We leverage this property in our analysis. In particular, let γ_t be such that $f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}$ for every constraint. Then, the log barrier is locally-smooth at point x_t with constant $M_2(x_t)$

$$M_2(x_t) \leq M_0 + 6\eta \sum_{i=1}^m \frac{M_i}{\alpha_t^i} + 20\eta \sum_{i=1}^m \frac{(\theta_t^i)^2}{(\alpha_t^i)^2}, \tag{4.8}$$

where $\theta_t^i = \langle \nabla f^i(x_t), \frac{g_t}{\|g_t\|} \rangle$, and $\alpha_t^i = -f^i(x_t)$ for all $i \in [m]$. The growth on the constraints can be bounded by any constant in $(0,1)$, we pick $\frac{1}{2}$ for simplicity, similarly to Hinder and Ye [HY18]. In more details, this adaptivity property and the $M_2(x_t)$-local smoothness are analyzed in Section 4.2.3. Importantly, our local smoothness $M_2(x)$ bound is more accurate since is constructed by using the smoothness of the constraints and takes into account the gradients measurements, in contrast to Hinder and Ye [HY19] whose bound is built using the Lipschitz continuity without the gradient measurements.

The log barrier gradient estimator

The key ingredient of the log barrier method together with the safe step size is estimating the log barrier gradient.

Estimating the gradient Recall that the log barrier gradient by definition is:

$$\nabla B_\eta(x_t) = \nabla f^0(x_t) + \eta \sum_{i=1}^{m} \frac{\nabla f^i(x_t)}{\alpha_t^i}.$$

Since we only have the stochastic information, we estimate the log barrier gradient as follows:

Algorithm 2 Log Barrier Gradient estimator $\mathcal{O}_\eta(x_t, n)$

```
1: Input: Oracles $F^i(\cdot,\xi), G^i(\cdot,\xi)$, $\forall i \in \{0,\dots,m\}$, $x_t \in \mathcal{X}$, number of measurements $n$
2: $g_t \leftarrow G_n^0(x_t,\xi_t) + \eta \sum_{i=1}^m \frac{G_n^i(x_t,\xi_t)}{-F_n^i(x_t,\xi_t)}$;
3: Output: $g_t$
```

In the above we allow to take a batch of measurements per call and average them as defined in Eq. (4.2) in order to reduce the variances $\sigma_i^2(n) := \frac{\sigma_i^2}{n}, \hat{\sigma}_i^2(n) := \frac{\hat{\sigma}_i^2}{n}$.

Properties of the estimator The log barrier gradient estimator defined above is biased and can be heavy tailed since a part of it is a ratio of two sub-Gaussian random variables. Therefore, in the following lemma we provide a general high-confidence upper bound on the deviation. We denote $\bar{\alpha}_t^i := -F_n^i(x_t, \xi_t)$.

Lemma 2. *The deviation of the log barrier gradient estimator $\Delta_t := g_t - \nabla B_\eta(x_t)$ satisfies:*

$$\mathbb{P}\left\{ \|\Delta_t\| \le \hat{b}_0 + \hat{\sigma}_0(n)\sqrt{\ln\frac{1}{\delta}} + \sum_{i=1}^m \frac{\eta}{\bar{\alpha}_t^i}\left(\hat{b}_i + \hat{\sigma}_i(n)\sqrt{\ln\frac{1}{\delta}}\right) + \sum_{i=1}^m L_i \frac{\eta\sigma_i(n)}{\alpha_t^i\bar{\alpha}_t^i}\sqrt{\ln\frac{1}{\delta}} \right\} \ge 1-\delta. \tag{4.9}$$

From the above bound we can see that the closer we are to the boundary, the smaller is α_t^i, and the smaller variance σ_i we require to keep the same level of disturbance of the the log barrier gradient estimator. That is, the closer to the boundary, the more measurements we require to stay safe despite the disturbance, which does actually make sense. For the proof see Section B.1.

The above deviation consists of the variance part and the bias part. Note that the bias is non-zero even if the biases of the gradient estimators are zero. It can be bounded as follows (see Section B.1):

$$\|\mathbb{E}\Delta_t\| \leq \sum_{i=1}^{m} \frac{\eta L_i \sigma_i(n)}{(\alpha_t^i)^2} + \hat{b}_0 + \sum_{i=1}^{m} \frac{\eta}{\alpha_t^i} \hat{b}_i. \tag{4.10}$$

In the above, the expectation is taken given the x_t is fixed. This bias comes from the fact that we are estimating the ratio of two sub-Gaussian distributions, which is often heavy-tailed and even for Gaussian variables might behave very badly if the mean of the denominator is smaller than its variance [DR13]. This fact influences the SGD analysis, and does not allow getting convergence guarantees with larger noise. Therefore, our algorithm is very sensitive to the noise $\sigma_i(n)$ and might require many samples per iteration to reduce this noise.

Adaptive step-size γ_t

First of all, recall that the log barrier is non-smooth in general sense on $\mathcal{X}$, since it grows to infinity on the boundary. However, we can use the notion of the $M_2(x_t)$-local smoothness, that guarantees smoothness in a bounded region around the current point x_t: $\{x_{t+1} \in \mathcal{X} | f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}, x_t \in \mathcal{X}\}$. *The local smoothness of the Log Barrier $B_\eta(x)$ is required for our convergence analysis of the SGD.*

$M_2(x_t)$-local smoothness constant for the log barrier We derive our local smoothness constant based on the M_i-smoothness of the objective and constraints f^i for $i = 0, \ldots, m$. Compared to the Lipschitz constant-based approach (used in Hinder and Ye [HY19] and Usmanova, Krause, and Kamgarpour [UKK20]), our way to bound the local smoothness constant $M_2(x_t)$ allows to estimate it more tightly using the constraint gradients measurements.

Lemma 3. *On the bounded area around x_t within the radius γ_t such that the next iterate is restricted by $f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}$, along the step direction g_t the log barrier $B_\eta(x_t)$ is locally-smooth with*

$$M_2(x_t) := M_0 + 6\eta \sum_{i=1}^{m} \frac{M_i}{\alpha_t^i} + 20\eta \sum_{i=1}^{m} \frac{(\theta_t^i)^2}{(\alpha_t^i)^2}, \tag{4.11}$$

where $\theta_t^i = \langle \nabla f^i(x_t), \frac{g_t}{\|g_t\|} \rangle$.

For the proof of Lemma 3 see Appendix B.3. In the case with inexact measurements, we have to use lower bounds on α_t^i and upper bounds on θ_t^i. We denote by $\underline{\alpha}_t^i := \bar{\alpha}_t^i - \sigma_i(n)\sqrt{\ln \frac{1}{\delta}}$ a lower bound on α_t^i : $\mathbb{P}\{\alpha_t^i \geq \underline{\alpha}_t^i\} \geq 1 - \delta$. We denote an upper bound on θ_t^i by $\hat{\theta}_t^i := |\langle G_n^i(x, \xi), \frac{g_t}{\|g_t\|} \rangle| + \hat{b}_i + \hat{\sigma}_i(n)\sqrt{\log \frac{1}{\delta}}$, $\forall i \in [m]$ such that $\mathbb{P}\{\theta_t^i \leq \hat{\theta}_t^i\} \geq 1 - \delta$. Then, an upper bound on $M_2(x_t)$ can be computed as follows

$$\hat{M}_2(x_t) = M_0 + 6\eta \sum_{i=1}^{m} \frac{M_i}{\underline{\alpha}_t^i} + 20\eta \sum_{i=1}^{m} \frac{(\hat{\theta}_t^i)^2}{(\underline{\alpha}_t^i)^2}. \tag{4.12}$$

Adaptivity of the step-size In the above, to bound the local smoothness of the log barrier at the next iterate $x_{t+1} = x_t - \gamma_t g_t$, we require to bound the step size γ_t by the value, restricting the next iterate by the area $\left\{x_{t+1} \in \mathcal{X} | f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}, x_t \in \mathcal{X}\right\}$. Automatically, such condition guarantees the feasibility of x_{t+1} given the feasibility of x_t.

One way to get the adaptive step size γ_t is to use the Lipschitz constants L_i of f^i to bound γ_t (see Hinder and Ye [HY19] and Usmanova, Krause, and Kamgarpour [UKK20]):

$$\gamma_t \leq \min_{i \in [m]} \frac{-f^i(x_t)}{2L_i} \frac{1}{\|g_t\|}.$$

In practice, L_i are typically unknown or overestimated. For example, even in the quadratic case $f^i(x) = \|x\|^2$, L_i depends on the diameter of the set $\mathcal{X}$, and thus might be very conservative in the middle of the set. Again, we propose to use the smoothness constants M_i for safety instead.[1] Of course, smoothness parameter also can be overestimated, and as a future work we consider incorporating the problem-adaptive techniques [VDB21] or efficient constants estimation [Faz+19].

Lemma 4. *The adaptive safe step size γ_t bounded by*

$$\gamma_t \leq \min_{i \in [m]} \left\{ \frac{\alpha_t^i}{2|\theta_t^i| + \sqrt{\alpha_t^i M_i}} \right\} \frac{1}{\|g_t\|},$$

guarantees $f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}$.

The proof is based on the smoothness bound on the constraint growth:

$$f^i(x_{t+1}) \leq f^i(x_t) - \gamma_t \langle \nabla f^i(x_t), g_t \rangle + \gamma_t^2 \frac{M_i}{2} \|g_t\|^2.$$

For the full proof see Section B.2. We illustrate the principle of choosing this adaptive bound on Figure 4.2. Then, finally, we set the step-size to:

$$\gamma_t = \min \left\{ \min_{i \in [m]} \left\{ \frac{\underline{\alpha}_t^i}{2|\hat{\theta}_t^i| + \sqrt{\underline{\alpha}_t^i M_i}} \right\} \frac{1}{\|g_t\|}, \frac{1}{\hat{M}_2(x_t)} \right\}. \tag{4.13}$$

[1] *Firstly,* the Lipschitz constant, even if it is tight, provides the first-order linear upper bound on the constraint growth, whereas using the smoothness constant we can exploit more reliable and tight second order upper bound on the constraint growth. *Secondly,* Lipschitz constant is often much harder to estimate since it might strongly depend on the size of the set. By the same reason, in practice, even for the hard functions modeled by a neural network with smooth activation functions, we can estimate the smoothness parameters, but it is much more unclear how to estimate the Lipschitz constants properly.

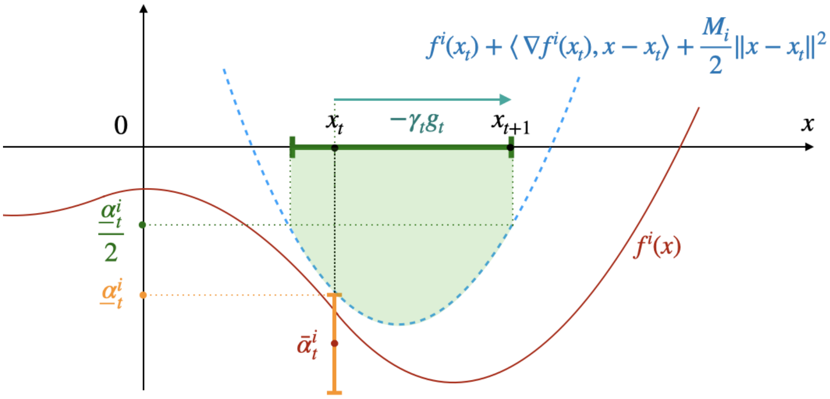

$f^i(x_t) + \langle \nabla f^i(x_t), x - x_t \rangle + \frac{M_i}{2}\|x - x_t\|^2$
0
x_t
$-\gamma_t g_t$
x_{t+1}
x
$\frac{\underline{\alpha}_t^i}{2}$
$\underline{\alpha}_t^i$
$\bar{\alpha}_t^i$
$f^i(x)$

Figure 4.2: Illustration of the adaptivity. Step size γ_t is chosen such that the quadratic smoothness upper bound (*blue*) on the constraint guarantees $f^i(x_{t+1}) \le f^i(x_t)/2$. $\underline{\alpha}_t^i$ is the lower bound on $\alpha_t^i = -f^i(x_t)$, constructed based on the mean estimator $\bar{\alpha}_t^i$. By the *orange* interval we denote the confidence interval for α_t^i. By the *green* interval we denote the adaptive region for x_{t+1} based on the requirement $f^i(x_{t+1}) \le f^i(x_t)/2$.

Basic algorithm

To sum up, below we propose our basic algorithm, but emphasize that it can be instantiated differently for different types of problems. We showcase possible instantiations of LB-SGD in following sections.

Algorithm 3 LB-SGD(x_0, T)

```
Input: $M_i > 0, i \in \{0, \dots, m\}, D > 0, \sigma_i, \hat{\sigma}_i, \hat{b}_i, n, T > 0, \delta > 0$;
for $t = 1, \dots, T$ do
    Set $g_t \leftarrow \mathcal{O}_\eta(x_t, n)$ by taking a batch of measurements of size $n$ at $x_t$;
    Compute lower bounds $\underline{\alpha}_t^i := \bar{\alpha}_t^i - \sigma_i(n)\sqrt{\ln\frac{1}{\delta}}$ , $\forall i \in [m]$;
    Compute upper bounds $\hat{\theta}_t^i = |\langle G_n^i(x, \xi), \frac{g_t}{\|g_t\|}\rangle| + \hat{b}_i + \hat{\sigma}_i(n)\sqrt{\log\frac{1}{\delta}}$, $\forall i \in [m]$;
    Compute $\hat{M}_2(x_t)$ using Eq. (4.11);
    $\gamma_t \leftarrow \min\left\{\min_{i\in[m]}\left\{\frac{\underline{\alpha}_t^i}{2|\hat{\theta}_t^i| + \sqrt{\underline{\alpha}_t^i M_i}}\right\}\frac{1}{\|g_t\|}, \frac{1}{\hat{M}_2(x_t)}\right\}$;
    $x_{t+1} \leftarrow x_t - \gamma_t g_t$;
end for
Output: $\{x_1, \dots, x_T\}$.
```

4.2.4 Safety

From the safety side, the adaptive step-size γ_t automatically guarantees the safety of all the iterates due to construction, for any procedure generating the iterations in the form $x_{t+1} = x_t - \gamma_t g_t$ where γ_t is bounded by Eq. (4.13). The feasibility of the optimization trajectory we guarantee with probability at least $1 - \hat{\delta}$ with $\hat{\delta} := mT\delta$.

Theorem 3. *Let $T > 0$ denote the total number of iterations of the form Eq. (4.7), and $\hat{\delta} \in (0, 1)$ denote the target confidence level. Then, for LB-SGD with parameter $\delta \leq \hat{\delta}/mT$, all the query points x_t are feasible with probability greater than $1 - \delta$.*

Proof. Due to the adaptive step size γ_t, we have $x_t \in \mathcal{X}$ implies $x_{t+1} \in \mathcal{X}$ (see Lemma 4) since $f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}$ with probability $1 - \delta$. Then, using $x_0 \in \mathcal{X}$ and Boole's inequality, we conclude that the whole optimization trajectory $\{x_t\}_{t\in[T]}$ is feasible with probability at least $1 - mT\delta \geq 1 - \hat{\delta}$. ∎

4.3 Method options and convergence analysis

First of all, let us show the following general property of the log barrier method, important for the further convergence analysis of any problem type that we discuss.

4.3.1 Keeping a distance away from the boundary

Imagine that $\underline{\alpha}_t^i$ becomes 0 for some iteration t during the learning. That would lead to $\gamma_t = 0$, and the algorithm will stop without converging, since there is no safe non-zero step-size. Moreover, the log barrier gradient at that point simply blows up. However, we can lower bound the step sizes γ_t if we can provide a lower bound on $\underline{\alpha}_t^i$ for all $t > 0$ during the learning with high probability:

Lemma 5. *If $\underline{\alpha}_t^i \geq c\eta$ for $c > 0$, then we have $\mathbb{P}\{\gamma_t \geq C\eta\} \geq 1 - \delta$ with C defined by*

$$C := \frac{c}{2L^2(1 + \frac{m}{c})} \frac{1}{\max\left\{10 + \frac{3Mc\eta}{L^2}, 1 + \sqrt{\frac{Mc\eta}{4L^2}}\right\}}. \tag{4.14}$$

The proof is shown in Appendix B.5.

Therefore, for convergence, we need to show that our algorithm's iterates x_t do not only stay inside the feasible set, but moreover keep a distance away from the boundary. Keeping distance is the key property, guaranteeing the regularity of the log barrier function in the sense of a bounded gradient norm, bounded local smoothness and bounded variance. For the exact information case without noise, the adaptive gradient descent on the log barrier is shown to converge without stating this property explicitly [HY18]. However, in the stochastic case, this property becomes crucial for establishing stable convergence. It guarantees that the method pushes the iterates x_t away from the boundary of the set $\mathcal{X}$ as soon as they come too close to the boundary. We formulate it below.

Lemma 6. *Let Assumptions 1 to 3 hold, Assumption 4 hold with $\rho \geq \eta$, and let $\hat{\sigma}_i(n) \leq \frac{\alpha_t^i L}{\eta\sqrt{\ln\frac{1}{\delta}}}$, $\hat{b}_i \leq \frac{\alpha_t^i L}{2\eta}$, and $\sigma_i(n) \leq \frac{(\alpha_t^i)^2}{2\eta\sqrt{\ln\frac{1}{\delta}}}$. Then, we can show that for all x_t for all iterations $t \in [T]$ generated by the optimization process $x_{t+1} = x_t - \gamma_t g_t$ the following holds: $\mathbb{P}\{\forall t \in$*

$[T]$ $\min_{i\in[m]} \alpha_t^i \geq c\eta\} \geq 1-\hat{\delta}$ with

$$c := \left(\frac{l}{4L(2m+1)}\right)^m,$$

where $l > 0$ is defined as in Assumption 4.

Proof. First, let us note the following fact demonstrating that the product of the smallest absolute constraint values is not decreasing if x_t is close enough to the boundary.

Fact 3. *Let Assumptions 1 to 3 hold, Assumption 4 hold with $\rho \geq \eta$, and let $\hat{\sigma}_i(n) \leq \frac{\alpha_t^i L}{\eta\sqrt{\ln\frac{1}{\delta}}}$, $\hat{b}_i \leq \frac{\alpha_t^i L}{2\eta}$, and $\sigma_i(n) \leq \frac{(\alpha_t^i)^2}{2\eta\sqrt{\ln\frac{1}{\delta}}}$. If at iteration t we have $\min_{i\in[m]} \alpha_t^i \leq \bar{c}\eta$ with $\bar{c} := \frac{l}{L}\frac{1}{2m+1}$, then, for the next iteration $t+1$ we get $\prod_{i\in\mathcal{I}} \alpha_{t+1}^i \geq \prod_{i\in\mathcal{I}} \alpha_t^i$ for any $\mathcal{I} : \mathcal{I}_t \subseteq \mathcal{I}$ with $\mathcal{I}_t := \{i \in [m] : \alpha_t^i \leq \eta\}$ with probability $1-\delta$.*

For the proof see Appendix B.4.

Note that if $\min_{i\in[m]} \alpha_t^i \geq \bar{c}\eta$ for all $t \geq 0$, then the statement of the Lemma holds automatically. Now, consider a consecutive set of steps $t = \{t_0, \ldots, t_k\}$ on whose $\min_{i\in[m]} \alpha_t^i \leq \bar{c}\eta$. By definition, and using the Fact 3, for any $t \in \{t_0, \ldots, t_k\}$ we have with probability $1-\delta$

$$\prod_{i\in\mathcal{I}_{t+1}} \alpha_{t+1}^i = \frac{\prod_{i\in\mathcal{I}_t\cup\mathcal{I}_{t+1}} \alpha_{t+1}^i}{\prod_{i\in\mathcal{I}_t\setminus\mathcal{I}_{t+1}} \alpha_{t+1}^i} \geq \frac{\prod_{i\in\mathcal{I}_t\cup\mathcal{I}_{t+1}} \alpha_t^i}{\prod_{i\in\mathcal{I}_t\setminus\mathcal{I}_{t+1}} \alpha_{t+1}^i}.$$

By induction, applying the above sequentially for all $t \in \{t_0, \ldots, t_k\}$ we can get

$$\prod_{i\in\mathcal{I}_{t_k}} \alpha_{t_k}^i \geq \frac{\prod_{i\in\mathcal{I}_{t_k}\cup\mathcal{I}_{t_{k-1}}\cup\ldots\cup\mathcal{I}_{t_0}} \alpha_{t_0}^i}{\prod_{i\in\mathcal{I}_{t_{k-1}}\setminus\mathcal{I}_{t_k}} \alpha_{t_k}^i \prod_{i\in\mathcal{I}_{t_{k-2}}\setminus(\mathcal{I}_{t_k}\cup\mathcal{I}_{t_{k-1}})} \alpha_{t_{k-1}}^i \cdots \prod_{i\in\mathcal{I}_{t_0}\setminus(\mathcal{I}_{t_k}\cup\ldots\cup\mathcal{I}_{t_1})} \alpha_{t_1}^i}$$

with probability $1-\hat{\delta}$ (using Boole's inequality).

Note that by definition of $\mathcal{I}_t$: $\alpha_t^i \leq \eta$ for $i \in \mathcal{I}_t$. At the same time, due to the step size choice, we have $\alpha_{t+1}^i \leq 2\alpha_t^i \leq 2\eta$. Also, note that the sum of the set cardinalities in the denominator equals to the cardinality of the set $|\mathcal{I}_{t_0} \cup \ldots \cup \mathcal{I}_{t_k} \setminus \mathcal{I}_{t_k}|$. Hence, with probability $1-\hat{\delta}$

$$\prod_{i\in\mathcal{I}_{t_k}} \alpha_{t_k}^i \geq \frac{\prod_{i\in\mathcal{I}_{t_k}\cup\mathcal{I}_{t_{k-1}}\cup\ldots\cup\mathcal{I}_{t_0}} \alpha_{t_0}^i}{(2\eta)^{|\mathcal{I}_{t_0}\cup\ldots\cup\mathcal{I}_{t_k}\setminus\mathcal{I}_{t_k}|}}.$$

Thus, for any $j \in \mathcal{I}_{t_k}$ we get the bound:

$$\alpha_{t_k}^j \geq \frac{\prod_{i\in\mathcal{I}_{t_k}\cup\mathcal{I}_{t_{k-1}}\cup\ldots\cup\mathcal{I}_{t_0}} \alpha_{t_0}^i}{(2\hat{c}\eta)^{|\mathcal{I}_{t_0}\cup\ldots\cup\mathcal{I}_{t_k}\setminus j|}} \geq \frac{(\bar{c}\eta/2)^{|\mathcal{I}_{t_0}\cup\ldots\cup\mathcal{I}_{t_k}|}}{2^{|\mathcal{I}_0\cup\ldots\cup\mathcal{I}_{t_k}|-1}} \geq \left(\frac{\bar{c}}{4}\right)^m \eta.$$

Using the definition $\bar{c} := \frac{l}{L}\frac{1}{2m+1}$, we obtain the statement of the lemma. ■

4.3.2 Stochastic non-convex problems

For the non-convex problem we analyse LB-SGD(x_0, T) with the fixed parameter η, that uses the stopping criterion $\|g_t\| \leq 3\eta/4$ and outputs $x_{\hat{t}}$ with $\hat{t}$ corresponding to $\arg\min_{t\in T} \|g_t\|$.

Stationarity criterion in the non-convex case

Similarly to Usmanova, Krause, and Kamgarpour [UKK20], we can state that in general case small gradient of the log barrier with parameter η leads to an η-approximate KKT point of the constrained problem. Let us set the pair of primal and dual variables to $(x, \lambda) := \left(x, \left[\frac{\eta}{-f^1(x)}, \dots, \frac{\eta}{-f^m(x)}\right]^T\right)$. Then, it satisfies:

$$1)\ \|\nabla_x \mathcal{L}(x, \lambda)\| = \|\nabla B_\eta(x)\|;$$
$$2)\ \lambda^i(-f^i(x)) = \frac{\eta}{-f^i(x)}(-f^i(x)) = \eta;$$
$$3)\ \lambda^i \geq 0, -f^i(x) \geq 0,\ i \in [m].$$

This insight immediately implies the following Lemma.

Lemma 7. *Consider problem (P) under Assumptions 1 to 3. Let $\hat{x}$ be an η-approximate solution to $\min_{x\in\mathbb{R}^d} B_\eta(x)$, the η-log barrier approximation of (P), such that $\|\nabla B_\eta(\hat{x})\| \leq \eta$. Then, $\hat{x}$ is an η-approximate KKT point to the original problem (P).*

Thus, the stationarity criterion for the general case is the small log barrier gradient norm.

Convergence for the non-convex problem

Then, we get the following convergence result:

Theorem 4. *After at most T iterations of LB-SGD with $T \leq 4\frac{B_\eta(x_0) - \min_x B_\eta(x)}{C\eta^3}$, and with $\sigma_i(n) = O(\frac{\eta^2}{D})$, $\hat{\sigma}_i(n) = O(\frac{\eta}{D})$, and $\hat{b}_i = O(\frac{\eta}{D})$, for the output $x_{\hat{t}}$ with $\hat{t} = \arg\min_{t\in T}\|g_t\|$ we have*

$$\mathbb{P}\left\{\|\nabla B_\eta(x_{\hat{t}})\| \leq \eta\right\} \geq 1 - \hat{\delta} \tag{4.15}$$

Therefore, given $\sigma_i(n) = \frac{\sigma_i}{\sqrt{n}}$ and $\hat{\sigma}_i(n) = \frac{\hat{\sigma}_i}{\sqrt{n}}$ Eq. (4.3), for constant $\hat{\sigma}_i, \sigma_i$, we require $n = O(\frac{1}{\eta^4})$ oracle calls per iteration, and $N = O(\frac{1}{\eta^7})$ calls of the first-order stochastic oracle in total. Using Lemma 7, we get that $x_{\hat{t}}$ is an ε-approximate KKT point to the original problem (P) with $\varepsilon = \eta$.

Remark *Lower bound in the unconstrained non-safe case.* In a well-known model where algorithms access smooth, non-convex functions through queries to an unbiased stochastic gradient oracle with bounded variance, Arjevani, Carmon, Duchi, Foster, Srebro, and Woodworth [Arj+19] prove that in the worst case any algorithm requires at least ε^{-4} queries to find an ε stationary point. Although, they allow d to depend on ε. Therefore, we "pay" extra ε^{-3} measurements for safety. From the methodology point of view, this happens due to the non-smoothness of the log-barrier on the boundary and the fact that the noise of the barrier gradient estimator is very sensitive to how close the iterates x_t are to the boundary.

Proof. First, let us denote $\hat{\gamma}_t := \gamma_t \|g_t\|$. At each iteration of Algorithm 4 with the fixed η the value of the logarithmic barrier decreases at least by the following value:

$$\begin{aligned} B_\eta(x_t) - B_\eta(x_{t+1}) &\overset{\textcircled{1}}{\geq} \gamma_t \left\langle \nabla B_\eta(x_t), g_t \right\rangle - \frac{1}{2} M_2(x_t) \gamma_t^2 \|g_t\|^2 \\ &= \hat{\gamma}_t \langle \Delta_t, \frac{g_t}{\|g_t\|} \rangle + \hat{\gamma}_t \left(1 - \frac{M_2(x_t)\gamma_t}{2} \right) \|g_t\| \\ &\overset{\textcircled{2}}{\geq} \frac{1}{2} \hat{\gamma}_t \|g_t\| - \hat{\gamma}_t \|\Delta_t\|. \end{aligned} \tag{4.16}$$

In the above, $M_2(x)$ is a local smoothness constant that we bound by Eq. (4.11). The first inequality $\textcircled{1}$ is due to the local smoothness of the barrier. $\textcircled{2}$ is due to the fact that $\mathbb{P}\{\gamma_t \leq \frac{1}{M_2(x_t)}\} \geq 1 - \delta$, given $x_t \in Int(\mathcal{X})$. Summing up the above inequalities Eq. (4.16) for $t \in [T]$, we obtain:

$$\sum_{t \in [T]} \hat{\gamma}_t \left(\frac{1}{2} \|g_t\| - \|\Delta_t\| \right) \leq B_\eta(x_0) - \min_{x \in X} B_\eta(x).$$

Recall that we stop the round as soon as we get $\|g_t\| \leq 3\eta/4$. Hence, for all T iterations $t \in [T]$ with $\|g_t\| \geq 3\eta/4$ we have $\hat{\gamma}_t \geq 0.75\eta\gamma_t$. Therefore, we get:

$$T_k \leq \frac{B^\eta(x_0) - \min_x B^\eta(x)}{0.75\eta \max_{t \in [T]} \{\gamma_t \left(0.5\|g_t\| - \|\Delta_t\|\right)\}}. \tag{4.17}$$

We have obtain the lower bound on the denominator. Using the result of Lemma 13 and Lemma 6, for all $t \in [T_k]$ we have $\gamma_t \geq C\eta$. Next, we have to upper bound on $\|\Delta_t\|$ with high probability. By Lemma 2 combined with Lemma 6, we have $\mathbb{P}\{\|\Delta_t\| \leq \frac{\eta}{4}, \forall t\} \geq 1 - \delta$ for

$$\begin{aligned} &\hat{\sigma}^0(n) \leq \frac{\eta}{4(2m+1)\sqrt{\ln \frac{1}{\delta}}}, \hat{\sigma}^i(n) \leq \frac{\alpha_t^i}{4(2m+1)\sqrt{\ln \frac{1}{\delta}}}, \\ &\hat{b}^0 \leq \frac{\eta}{4(2m+1)}, \hat{b}^i \leq \frac{\alpha_t^i}{4(2m+1)}, \end{aligned}$$

$$\sigma^i(n) \leq \frac{(\underline{\alpha}_t^i)^2}{4(2m+1)L\sqrt{\ln\frac{1}{\delta}}},$$

(using the Boolean inequality). Combining it with inequality (4.17), the algorithm stops after at most T iterations with

$$T \leq 8\frac{B_\eta(x_0) - \min_x B_\eta(x)}{C\eta^3}.$$

Since $\mathbb{P}\{\|\Delta_t\| \leq \frac{\eta}{4}\forall t\} \geq 1 - \hat{\delta}$ and $\|g_t\| \geq 3\eta/4$, we obtain

$$\mathbb{P}\left\{\|\nabla B_\eta(x_{\hat{t}})\| \leq \eta\right\} \geq 1 - \hat{\delta}.$$

■

4.3.3 Stochastic convex problems

For the convex case, we propose to use LB-SGD(x_0, T) with the *output:* $\bar{x}_T := \frac{\sum_{t=1}^T \gamma_t x_t}{\sum_{t=1}^T \gamma_t}$. Next, we discuss the optimality criterion for convex problems.

Optimality criterion in the convex case

In the convex case, we can relate an approximate solution of the log barrier problem to an ε-approximate solution of the original problem in terms of the objective value.

Assumption 5. *The objective and the constraint functions $f^i(x)$ for all $i \in \{0, \dots, m\}$ are convex.*

Note that Assumption 3 implies non-emptiness on $Int(\mathcal{X})$ which is called Slater Constraint Qualification. In the convex setting, it in turn implies the extended MFCQ:

Fact 4. *Let Assumptions 1, 3 and 5 hold. Then, Assumption 4 holds with $s_x := \frac{x - x_0}{\|x - x_0\|}$, such that $\langle \nabla f^i(x), s_x\rangle \geq \frac{\beta-\rho}{D}$ for all $i \in \mathcal{I}_\rho(x)$ for any $0 < \rho < \beta$.*

Proof. Indeed, for any point $x \in \mathcal{X}$ and for any convex constraint f^i such that $f^i(x) \geq -\rho$, due to convexity we have $f^i(x) - f^i(x_0) \leq \langle \nabla f^i(x), x - x_0\rangle$. Given the bounded diameter of the set $\|x_0 - x\| \leq D$, we get $\langle \nabla f^i(x), s_x\rangle \geq \frac{\beta-\rho}{D}$. ■

Then, we can relate an η-approximate solution by the log barrier value with an ε-approximate solution for the original problem, where ε depends on η linearly up to a logarithmic factor. We formulate that in the following lemma:

Lemma 8. *Consider problem (P) under Assumptions 1 to 3, and the convexity Assumption 5. Assume that $\hat{x}$ is an η-approximate solution to the η-log barrier approximation, that is,*

$$B_\eta(\hat{x}) - B_\eta(x^*_\eta) \leq \eta,$$

where x^*_η is a solution of the $\min B_\eta$ *minimization problem, with $\eta \leq \beta/2$. Then, $\hat{x}$ is an ε-approximate solution to the original problem (P) with $\varepsilon = \eta(m+1) + \eta m \log\left(\frac{2mLD\hat{\beta}}{\eta\beta}\right)$, that is, $f^0(\hat{x}) - \min_{x\in\mathcal{X}} f^0(x) \leq \varepsilon$, where $\hat{\beta} > 0$ is such that $\forall i \in [m] \forall x \in \mathcal{X}$ $|f^i(x)| \leq \hat{\beta}$. Since the constraints are smooth and the set $\mathcal{X}$ is bounded, such $\hat{\beta}$ exists.*

Proof sketch Let $\hat{x}$ be an approximately optimal point for the log barrier: $B_\eta(\hat{x}) - B_\eta(x^*_\eta) \leq \eta$, and x^*_η be an optimal point for the log barrier. Then, using the definition, we can bound: $f^0(\hat{x}) - f^0(x^*_\eta) \leq \eta + \eta \sum_{i=1}^m -\log \frac{-f^i(x^*_\eta)}{-f^i(\hat{x})}$. Combining Fact 4 with the first order stationarity criterion, we can derive: $\min_{i\in[m]}\{-f^i(x^*_\eta)\} \geq \frac{\eta\beta}{2mLD}$. Hence, combining the above two inequalities, we get the following relation of point $\hat{x}$ and point x^*_η: $f^0(\hat{x}) - f^0(x^*_\eta) \leq \eta\left(1 + m\log\left(\frac{2mLD\hat{\beta}}{\eta\beta}\right)\right)$ using $-f^i(\hat{x}) \leq \hat{\beta}$. Using the Lagrangian definition for stationarity of the optimal point of the initial problem x^*, we get the following relation between x^* and x^*_η: $f^0(x^*_\eta) - f^0(x^*) \leq m\eta$. Combining it with the above, we get the statement of the Lemma

$$f^0(\hat{x}) - \min_{x\in\mathcal{X}} f^0(x) \leq \eta + \eta m \log\left(\frac{2mLD\hat{\beta}}{\eta\beta}\right) + m\eta.$$

For the full proof see Appendix B.6.

Convergence in the convex case

As already discussed in the optimality criterion Section 4.3.3, for the convex problem we only require the convergence in terms of the value of the log barrier. Thus, we get the following convergence result for this method.

Theorem 5. *Let $B_\eta(x) := f^0(x) - \eta\sum_{i=1}^m \log(-f^i(x))$ be a log barrier function with parameter $\eta > 0$, and $x_0 \in \mathbb{R}^d$ be the starting point. Let x^*_η be a minimizer of $B_\eta(x)$. Then, after $T \geq \frac{\|x_0 - x^*\|^2}{C\eta^2}$ iterations of LB-SGD, and with $\sigma_i(n) = O(\frac{\eta^2}{D})$, $\hat{\sigma}_i(n) = O(\frac{\eta}{D})$, and $\hat{b}_i = O(\frac{\eta}{D})$, for the point $\bar{x}_T := \frac{\sum_{t=1}^T \gamma_t x_t}{\sum_{t=1}^T \gamma_t}$ we obtain:*

$$\mathbb{P}\left\{B_\eta(\bar{x}_T) - B_\eta(x^*_\eta) \leq \eta\right\} \geq 1 - \hat{\delta}.$$

For the noise with constant variances $\hat{\sigma}_i, \sigma_i$, given $\sigma_i(n) = \frac{\sigma_i}{\sqrt{n}}$ Eq. (4.3) and $\hat{\sigma}_i(n) = \frac{\hat{\sigma}_i}{\sqrt{n}}$ Eq. (4.4), we require $n = O(\frac{1}{\eta^4})$ oracle calls per iteration, and $O(\frac{1}{\eta^6})$ measurements of the first-order oracle in total. Using Lemma 8 we get $\bar{x}_T$ is an ε-approximate solution to the original problem (P) with $\varepsilon = \eta(m+1) + \eta m \log\left(\frac{2mLD\hat{\beta}}{\eta\beta}\right)$, that is, $f^0(\hat{x}) - \min_{x\in\mathcal{X}} f^0(x) \leq \varepsilon$.

Proof. Note the following

$$B_\eta(x_{t+1}) - B_\eta(x^*_\eta) \overset{\textcircled{1}}{\leq} B_\eta(x_t) + \langle \nabla B_\eta(x_t), x_{t+1} - x_t\rangle + \frac{M_2(x_t)}{2}\|x_t - x_{t+1}\|^2 - B_\eta(x^*_\eta)$$

$$
\begin{aligned}
&\overset{\textcircled{2}}{\leq} \langle \nabla B_\eta(x_t), x_{t+1} - x_t \rangle + \frac{M_2(x_t)}{2} \|x_t - x_{t+1}\|^2 + \langle \nabla B_\eta(x_t), x_t - x_\eta^* \rangle \\
&= \frac{M_2(x_t)}{2} \|x_t - x_{t+1}\|^2 + \langle g_t, x_{t+1} - x_\eta^* \rangle - \langle \Delta_t, x_{t+1} - x_\eta^* \rangle \\
&\overset{\textcircled{3}}{\leq} \frac{\|x_t - x_\eta^*\|^2}{2\gamma_t} - \frac{\|x_{t+1} - x_\eta^*\|^2}{2\gamma_t} - \left(\frac{1}{2\gamma_t} - \frac{M_2(x_t)}{2} \right) \|x_{t+1} - x_t\|^2 - \langle \Delta_t, x_{t+1} - x_\eta^* \rangle \\
&\overset{\textcircled{4}}{\leq} \frac{\|x_t - x_\eta^*\|^2}{2\gamma_t} - \frac{\|x_{t+1} - x_\eta^*\|^2}{2\gamma_t} - \langle \Delta_t, x_{t+1} - x_\eta^* \rangle.
\end{aligned}
$$

The first inequality $\textcircled{1}$ is due to the $M_2(x_t)$-local smoothness of the log barrier, the second one $\textcircled{2}$ is due to convexity. The third inequality $\textcircled{3}$ uses the fact that: $\forall u \in \mathbb{R}^d : \ \langle g_t, x_{t+1} - u \rangle = \frac{\|x_t - u\|^2}{2\gamma_t} - \frac{\|x_{t+1} - u\|^2}{2\gamma_t} - \frac{\|x_{t+1} - x_t\|^2}{2\gamma_t}$. And the last one $\textcircled{4}$ is due to $\gamma_t \leq \frac{1}{M_2(x_t)}$. By multiplying both sides by γ_t, we get:

$$
2\gamma_t (B_\eta(x_{t+1}) - B_\eta(x_\eta^*)) \leq \|x_t - x_\eta^*\|^2 - \|x_{t+1} - x_\eta^*\|^2 - 2\gamma_t \langle \Delta_t, x_{t+1} - x_\eta^* \rangle. \tag{4.18}
$$

Then, by summing up the above for all $t \in [T]$ we get, and using the Jensen's inequality:

$$
\begin{aligned}
B_\eta \left(\frac{\sum_{t=1}^T \gamma_t x_t}{\sum \gamma_t} \right) - B_\eta(x_\eta^*) &\leq \frac{1}{\sum_{t=1}^T \gamma_t} \sum_{t=1}^T \gamma_t (B_\eta(x_t) - B_\eta(x_\eta^*)) \\
&\leq \frac{1}{2\sum_{t=1}^T \gamma_t} \sum_{t=1}^T \left(\|x_t - x_\eta^*\|^2 - \|x_{t+1} - x_\eta^*\|^2 - 2\gamma_t \langle \Delta_t, x_{t+1} - x_\eta^* \rangle \right) \\
&\leq \frac{\|x_0 - x_\eta^*\|^2}{2\sum_{t=1}^T \gamma_t} - \frac{\sum_{t=1}^T \gamma_t \langle \Delta_t, x_{t+1} - x_\eta^* \rangle}{2\sum_{t=1}^T \gamma_t}.
\end{aligned} \tag{4.19}
$$

That is, we can bound the accuracy by

$$
B_\eta(\bar{x}_T) - B_\eta(x_\eta^*) \leq \frac{D^2}{2\sum_{t=1}^T \gamma_t} + \frac{\max_t \langle \Delta_t, x_{t+1} - x_\eta^* \rangle}{2}. \tag{4.20}
$$

Using Lemma 13 we can prove for $\hat{\sigma}_i(n) \leq \frac{L\alpha_t^i}{3\eta\sqrt{\ln \frac{1}{\delta}}}$ that $\gamma_t \geq C\eta$. Recall from Lemma 2 Eq. (4.9):

$$
\mathbb{P}\left\{ \|\Delta_t\| \leq b_0 + \hat{\sigma}_0(n)\sqrt{\ln \frac{1}{\delta}} + \sum_{i=1}^m \frac{\eta}{\bar{\alpha}_t^i} \left(\hat{b}_i + \hat{\sigma}_i(n)\sqrt{\ln \frac{1}{\delta}} \right) + \sum_{i=1}^m L_i \frac{\eta\sigma_i(n)}{\alpha_t^i \bar{\alpha}_t^i} \sqrt{\ln \frac{1}{\delta}} \right\} \geq 1 - \delta. \tag{4.21}
$$

Hence, we can guarantee $\|\Delta_t\| \leq \frac{\eta}{2D}$ for all $t \in [T]$ with probability $1 - \hat{\delta}$ if for all $i \in [m]$

$$
\hat{\sigma}_0(n) \leq \frac{\eta}{2(2m+1)D}, \hat{\sigma}_i(n) \leq \frac{\alpha_t^i}{2(2m+1)D\sqrt{\ln \frac{1}{\delta}}}, \tag{4.22}
$$

$$\hat{b}_i \leq \frac{\alpha_t^i}{2(2m+1)D}, \sigma_i(n) \leq \frac{(\alpha_t^i)^2}{2(2m+1)LD\sqrt{\ln\frac{1}{\delta}}}. \tag{4.23}$$

Therefore, we get for the $\bar{x}_T$ the following bound on the accuracy:

$$B_\eta(\bar{x}_T) - B_\eta(x_\eta^*) \leq \frac{\|x_0 - x_\eta^*\|^2}{TC\eta} + \max\|\Delta_t\|D \leq \frac{\|x_0 - x_\eta^*\|^2}{TC\eta} + \frac{\eta}{2}. \tag{4.24}$$

Thus, for $T \geq \frac{D^2}{2C\eta^2}$ we obtain $\mathbb{P}\{B(\bar{x}_T) - B(x^*) \leq \eta\} \geq 1 - \hat{\delta}$.

In order to satisfy conditions on the variance Eq. (4.22), we require at each iteration $n = O\left(\frac{1}{\eta^4}\right)$ measurements, and therefore $N = Tn = O(\frac{1}{\eta^6})$ measurements in total. ∎

4.3.4 Strongly-convex problems

For the strongly convex case, we make use of restarts with iteratively decreasing parameter η:

Algorithm 4 LB-SGD with decreasing η $(\eta_0, \eta > 0, x_0 \in \mathbb{R}^d, \omega \in (0,1), \{T_k\})$

```
1: Input: $M_i > 0, i \in \{0,\dots,m\}, \eta_0$, $\hat{x}_0 \leftarrow x_0$, $K = \log_2\frac{\eta_0}{\eta}$;
2: for $k = 0,\dots,K-1$ do
3:     $\hat{x}_{k+1} \leftarrow$ LB-SGD$(\hat{x}_k, T_k)$;
4:     $\eta_{k+1} \leftarrow \omega\eta_k$, with $\omega \in (0,1)$;
5: end for
6: Output: $\hat{x}_K$.
```

Convergence

Theorem 6. *Let $B_\eta(x) := f^0(x) - \eta\sum_{i=1}^m \log(-f^i(x))$ be a μ-strongly-convex log barrier function with parameter η, $x_0 \in \mathbb{R}^d$ be the starting point. Then, after at most $T = \frac{\|x_0 - x_\eta^*\|^2}{C\eta_0^2} + \sum_{k=1}^K O(\frac{1}{C\mu\eta_k}) = O\left(\frac{\ln\frac{\eta_0}{\eta}}{\mu\eta}\right)$ iterations of LB-SGD with decreasing η, and with $\sigma_i(n) = O(\frac{\eta^2}{mL_i})$, $\hat{b}_i = O(\eta M_i)$, and $\hat{\sigma}_i(n) = O(\eta M_i)$, we obtain:*

$$\mathbb{P}\left\{B_\eta(\hat{x}_K) - \min_{x\in\mathcal{X}} B_\eta(x) \leq \eta\right\} \geq 1 - \hat{\delta}$$

Hence, for the constant $\hat{\sigma}_i, \sigma_i$, we require $n = O(\frac{1}{\eta^4})$ measurements per iteration, and $N = O(\frac{1}{\eta^5})$ measurements of the first-order oracle in total. Using Lemma 8, we obtain that $\hat{x}_K$ is an ε-approximate solution to the original problem (P) with $\varepsilon = \eta(m+1) + \eta m \log\left(\frac{2mLD\hat{\beta}}{\eta\beta}\right)$.

Proof. Let $x_{\eta_k}^*$ be the unique minimizer of $B_{\eta_k}(x)$. We do the restarts with decreasing

$\eta_{k+1} = \omega\eta_k$. From strong convexity, for $k > 0$ we have:

$$\|\hat{x}_{k-1} - x^*_{\eta_k}\|^2 \leq D_k^2 := \frac{B_{\eta_k}(\hat{x}_{k-1}) - B_{\eta_k}(x^*_{\eta_k})}{\mu}. \tag{4.25}$$

Moreover, from Eq. (4.16) with high probability $B_{\eta_k}(x_t) \leq B_{\eta_k}(x_{t-1})$ holds for any $t \in [T_k]$ if

$$\begin{aligned}
&\hat{\sigma}_0(n_k) \leq \frac{\eta_k}{4(2m+1)\sqrt{\ln\frac{1}{\delta}}}, \hat{\sigma}^i(n_k) \leq \frac{c\eta_k}{4(2m+1)\sqrt{\ln\frac{1}{\delta}}},\\
&\hat{b}^i \leq \frac{c\eta_k}{4(2m+1)}, \sigma^i(n_k) \leq \frac{c^2\eta_k^2}{4(2m+1)L\sqrt{\ln\frac{1}{\delta}}}.
\end{aligned}$$

Hence, by induction, we can bound:

$$\|\hat{x}_t - x^*_{\eta_k}\|^2 \leq \frac{B_{\eta_k}(x_t) - B_{\eta_k}(x^*_{\eta_k})}{\mu} \leq \frac{B_{\eta_k}(x_{t-1}) - B_{\eta_k}(x^*_{\eta_k})}{\mu} \leq \ldots \leq D_k^2. \tag{4.26}$$

Note that for all $x \in \mathcal{X}$ we have $B_{\eta_k}(x) \geq B_{\eta_{k-1}}(x)$ (without loss of generality assuming $-f^i(x) \leq \hat{\beta} \leq 1$). Consequently, $-B_{\eta_k}(x^*_{\eta_k}) \leq -B_{\eta_{k-1}}(x^*_{\eta_{k-1}})$. Therefore, using the definition of the log barrier, we can get:

$$\begin{aligned}
B_{\eta_k}(\hat{x}_{k-1}) - B_{\eta_k}(x^*_{\eta_k}) &\leq B_{\eta_{k-1}}(\hat{x}_{k-1}) - B_{\eta_{k-1}}(x^*_{\eta_{k-1}}) + (\eta_{k-1} - \eta_k)\sum_{i=1}^m -\log -f^i(\hat{x}_{k-1})\\
&\leq B_{\eta_{k-1}}(\hat{x}_{k-1}) - B_{\eta_{k-1}}(x^*_{\eta_{k-1}}) - m(\eta_{k-1} - \eta_k)\log c\eta_k. \qquad (4.27)
\end{aligned}$$

As a base of induction we assume that for $k-1$, we have $B_{\eta_{k-1}}(\hat{x}_{k-1}) - B_{\eta_{k-1}}(x^*_{\eta_k}) \leq \eta_{k-1}$. Combining the inequalities Eq. (A.11) and Eq. (4.27), we get:

$$D_k^2 \leq \frac{\eta_{k-1} + m(\eta_{k-1} - \eta_k)\log\frac{1}{c\eta_k}}{\mu} \leq \frac{\omega^{-1}\eta_k(1 + (1-\omega)m\log\frac{1}{c\eta_k})}{\mu}. \tag{4.28}$$

From the previous Theorem 5, for $k > 0$ we have $B_{\eta_k}(\hat{x}_k) - B_{\eta_k}(x^*_{\eta_k}) \leq \eta_k$, for $T_k = \frac{D_k^2}{C\eta_k^2}$, $\sigma_i(n_k) \leq \frac{\eta_k^2}{D_k\sqrt{\ln\frac{1}{\delta}}}$, and $\hat{\sigma}_i(n_k) \leq \frac{\eta_k}{D_k}\frac{1}{\sqrt{\ln 1/\delta}}$. Inserting the result of Eq. (4.28) into these bounds, we require for $k > 0$: $T_k = \frac{\omega^{-1}+(\omega^{-1}-1)m\log\frac{1}{c\eta_k}}{\mu C\eta_k}$, $\sigma_i(n_k) \leq \frac{1}{\sqrt{\ln 1/\delta}}\min\{\frac{\mu\eta_k^{1.5}}{2}, \sqrt{C}\eta_k, \frac{c^2\eta_k^2}{mL_i}\}$, and $\hat{\sigma}_i(n_k) \leq \frac{1}{\sqrt{\ln 1/\delta}}\min\{\frac{\mu}{2}, \frac{M\sqrt{C}\eta_k}{2}, \frac{c\eta_k}{mL_i}\}$. That is, in total we need the following number of iterations

$$\begin{aligned}
T = T_0 + \sum_{k=1}^K T_k &= \frac{R^2}{C\eta_0^2} + \sum_{k=1}^K \frac{\omega^{-1} + m(\omega^{-1} - 1)\log\frac{1}{c\eta_k}}{\mu C\eta_k}\\
&\leq \frac{D^2}{C\eta_0^2} + \frac{\omega^{-1} + m(\omega^{-1} - 1)\log\frac{1}{c\eta}}{\mu C\eta}\log\frac{\eta_0}{\eta} = \tilde{O}\left(\frac{1}{\mu\eta}\right).
\end{aligned}$$

And the number of measurements in the case of the noise with constant variances in total is:

$$
\begin{aligned}
T = T_0 n_0 + \sum_{k=1}^{K} T_k n_k &\leq \frac{D^2}{C\eta_0^6} + \sum_{k=1}^{K} \frac{\omega^{-1} + m(\omega^{-1} - 1)\log\frac{1}{c\eta_k}}{\mu C \eta_k} \frac{1}{\eta_k^4} \\
&\leq \frac{D^2}{\mu C\eta_0^6} + \frac{\omega^{-1} + m(\omega^{-1} - 1)\log\frac{1}{c\eta}}{\mu C \eta^5} \log\frac{\eta_0}{\eta} = \tilde{O}\left(\frac{1}{\mu\eta^5}\right).
\end{aligned}
$$

■

4.4 Zeroth-order optimization

A special case of stochastic optimization is zeroth-order optimization, in which one can access only the value measurements of f^i. In many applications, for example in physical systems with measurements collected by noisy sensors, we only have access to noisy evaluations of the functions. Formally we assume access to a *one-point stochastic zeroth-order oracle*, as defined in Eq. (2.3). That is, for any $i \in \{0, \dots, m\}$ this oracle provides noisy function evaluations at the requested point x_j: $F^i(x_j, \xi_j^i) = f^i(x_j) + \xi_j^i$, where ξ_j^i is a zero-mean σ_i^2-sub-Gaussian noise.

Zeroth-order gradient estimator One way to tackle zeroth-order optimization is to sample a random point $x_t + \nu s_t$ around x_t at iteration t, and approximate the stochastic gradient $G^i(x, \xi)$ using finite differences. A classical choice of the sampling distribution is the Gaussian distribution, referred to as Gaussian sampling. However, since the Gaussian distribution has infinite support, one has an additional risk of sampling a point in the unsafe region arbitrarily far from the point, which is inappropriate for *safe learning.* Therefore, we propose to use the uniform distribution $\mathcal{U}(\mathcal{S}^d)$ on the unit sphere for sampling. In particular, in the case where we only have access to a noisy zeroth-order oracle, we estimate the gradient in the following way.

We need to estimate the descent directions of f^i using the zeroth-order information. For any point x, we can estimate the gradient of the function ∇f^i by sampling directions s_j uniformly at random on the unit sphere $s_j \sim \mathcal{U}(\mathcal{S}^d)$, and using the finite difference as follows:

$$
G^i_{\nu,n}(x, \xi) := \frac{d}{n} \sum_{j=1}^{n} \frac{F^i(x + \nu s_j, \xi_j^{i+}) - F^i(x, \xi_j^{i-})}{\nu} s_j, \tag{4.29}
$$

where $\xi_j^{i\pm}$ are sampled from σ_i-sub-Gaussian distribution. Note that s_j also satisfy the sub-Gaussian condition. [2]

[2] There is also an option of using the one-point estimator $G^i_{\nu,n}(x, \xi) := \frac{d}{n}\sum_{j=1}^{n} \frac{F^i(x+\nu s_j, \xi_j^{i+})}{\nu} s_j$, but the variance of this estimator might be much higher. Note that even with zero-noise its variance grows

There are also several other ways to sample directions to estimate the gradient from finite-differences. Berahas, Cao, Choromanski, and Scheinberg [Ber+21] compared various zeroth-order gradient approximation methods and showed that their sample complexity has a similar dependence on the dimensionality d required for a precise gradient approximation. Deterministic coordinate sampling requires fewer samples due to smaller constants. However, we stick with sampling on the sphere because deterministic coordinate sampling requires the number of samples to be divisible by d. We want to keep flexibility on how many samples we can take per iteration; this number might be provided by the application. However, we note that any other sampling procedure can also be used.

Then, the estimator $G^i_{\nu,n}(x,\xi)$ defined above is a biased estimator of the gradient $\nabla f^i(x)$ and an unbiased estimator of the smoothed function gradient $\nabla f^i_\nu(x)$. The smoothed approximation f^i_ν of each function f^i is defined as follows:

Definition 1. *The ν-smoothed approximation of the function $f(x)$ is defined by $f_\nu(x) := \mathbb{E}_b f(x+\nu b)$, where b is uniformly distributed in the unit ball $\mathcal{B}^d$, and $\nu \geq 0$ is the sampling radius.*

Algorithm 5 Zeroth-order gradient-value estimator $(F^i_n(x,\xi), G^i_{\nu,n}(x,\xi))$

```
1: Input: $F^i(\cdot,\xi), i \in \{0,\dots,m\}$, $x \in \mathcal{X}$, $\nu > 0$, $n \in \mathbb{N}$;
2: Sample $n$ directions $s_j \sim \mathcal{U}(\mathcal{S}^d)$, sample $F^i(x+\nu s_j, \xi^{i+}_j)$ and $F^i(x, \xi^{i-}_j)$, $j \in [n]$;
3: Output:
```

$$F^i_n(x,\xi) := \frac{\sum_{j=1}^n F^i(x,\xi^{i-}_j)}{n}$$

$$G^i_{\nu,n}(x,\xi) := \frac{d}{n}\sum_{j=1}^n \frac{F^i(x+\nu s_j,\xi^{i+}_j) - F^i(x,\xi^{i-}_j)}{\nu} s_j$$

Lemma 9. *Let $f^i_\nu(x)$ be the ν-smoothed approximation of $f^i(x)$. Then $\mathbb{E}G^i_{\nu,n}(x,\xi) = \nabla f^i_\nu(x)$, where the expectation is taken over both s_j and $\xi^{i\pm}_j$ for all $j \in [n]$.*

Proof. First note that $\mathbb{E}G^i_{\nu,n}(x,\xi) = \underbrace{\mathbb{E}\frac{d}{n}\sum_{j=1}^n \frac{f^i(x+\nu s_j) - f^i(x)}{\nu} s_j}_{(1)} + \underbrace{\mathbb{E}\frac{d}{n}\sum_{j=1}^n \frac{\xi^{i+}_j - \xi^{i-}_j}{\nu} s_j}_{(2)}.$

Recall that $\xi^{i\pm}_j$ are independent on s_j and zero-mean, hence $(2) = 0$. The proof that $(1) = \nabla f^i_\nu(x)$ is classical [FKM05] and is based on Stokes' theorem. ■

The following lemma shows important properties of the above zeroth-order gradient-value estimators.

to infinity while $\nu \to 0$. Its variance would depend on $\frac{\max_{x\in\mathcal{X}} |f^i(x)|}{\nu}$, while the two-point estimator's variance depends on the Lipschitz constant L_i, which might be significantly smaller. Also, in the case of differentiable f^i with small noise ξ the two-point estimator becomes a finite difference directional derivative estimator with the accuracy dependent on ν only, in contrast to the one-point estimator.

Lemma 10. *Let $F^i(x,\xi)$ have variance $\sigma_i > 0$ and let the estimator $G^i_{\nu,n}(x,\xi)$ be defined as in (4.29) by sampling s_j uniformly from the unit sphere $\mathcal{U}(\mathcal{S}^d)$, then $f^i_\nu(x)$, and $G^i_{\nu,n}(x,\xi)$ are biased approximations of $f^i(x)$ and $\nabla f^i(x)$ respectively, such that*

$$|f^i(x) - f^i_\nu(x)| \leq \nu^2 M_i,$$

the variance of $F^i_n(x,\xi)$ is $\sigma_i(n) = \frac{\sigma_i}{\sqrt{n}}$ and the bias of $G^i_{\nu,n}(x,\xi)$ is bounded by:

$$\hat{b}_i := \|\nabla f^i(x) - \nabla f^i_\nu(x)\| \leq \nu M_i,\ \forall i \in \{0,\dots,m\}. \tag{4.30}$$

The variance of $G^i_{\nu,n}(x,\xi)$ is bounded as follows:

$$\hat{\sigma}_i^2(n) := \mathbb{E}\|G^i_{\nu,n}(x,\xi) - \nabla f^i_\nu(x)\|^2 \leq \frac{3}{n}\left(d\|\nabla f^i(x)\|^2 + \frac{d^2 M_i^2 \nu^2}{4}\right) + 4\frac{d^2}{n}\frac{\sigma_i^2}{\nu^2}\ \forall i \in \{0,\dots,m\}. \tag{4.31}$$

Proof. These properties are corollaries from Berahas, Cao, Choromanski, and Scheinberg [Ber+21]. For the bias (4.30) we use the result of Equation (2.35) [Ber+21], and for the variance (4.31) the result of Lemma 2.10 of the same paper, in both cases by setting the disturbance $\epsilon_f = 0$ in [Ber+21]. The last term of the variance is coming from the additive noise. We set the disturbance ϵ_f to zero for their formulation and analyze the noise separately since they consider the disturbance without any assumptions on it. In contrast, we consider the zero-mean and sub-Gaussian noise which we can use explicitly. For further discussions and proof, see Section B.7. ■

Setting the sample radius ν and bounding the sample complexity The parameters of the estimator defined in Algorithm 5 that we can control are ν and n. We want to set them in such a way that the biases b_i and variances $\sigma_i, \hat{\sigma}_i$ satisfy requirements of Theorems 4 to 6. Based on them, we can bound the sample complexity of our approach for zeroth-order setting.

According to Theorems 4 to 6, we require the bias to be bounded by $\hat{b}_i \leq \frac{\alpha_t^i}{2(2m+1)D}$, $\hat{b}_0 \leq \frac{\eta}{2(2m+1)D}$. Therefore, since $\hat{b}_i \leq \nu M_i$, we need to set the sampling radius small enough $\nu \leq \min\left\{\frac{\alpha_t^i}{2mM_iD}, \frac{\eta}{2mM_0D}\right\}$. Moreover, in order to guarantee *safety* of all the measurements within the sample radius ν around the current point $f^i(x_t + \nu s_t) \leq 0$ using the smoothness of each constraint

$$f^i(x_t + \nu s_t) \leq f^i(x_t) - \nu\langle\nabla f^i(x_t), s_t\rangle + \nu^2\frac{M_i}{2}\|s_t\|^2,$$

we require the sample radius to be $\nu \leq \frac{\alpha_t^i}{2\|\nabla f^i(x_t)\| + \sqrt{\alpha_t^i M_i}}$. This bound can be obtained using the same derivations as for the adaptive step size γ_t (Lemma 4). Hence, we set

$$\nu = \min\left\{\frac{\alpha_t^i}{2\|\nabla f^i(x_t)\| + \sqrt{\alpha_t^i M_i}}, \frac{\alpha_t^i}{2mM_iD}, \frac{\eta}{2mM_0}\right\} = O(\eta) = \Omega(\eta).$$

Thus, from the above Lemma 10 Eq. (4.31), the variance of the estimated gradient with $\nu = O(\varepsilon) = \Omega(\varepsilon)$ is

$$\hat{\sigma}_i^2(n) = \frac{1}{n} O\left(\max\left\{\frac{d^2\sigma_i^2}{\varepsilon^2}, L_i^2, d^2M_i^2\varepsilon^2\right\}\right). \tag{4.32}$$

Additionally, according to the previous Theorems 4 to 6, we require the variances to be $\hat{\sigma}_i(n) = O(\varepsilon)$ and $\sigma_i(n) = O(\varepsilon^2)$. From the above Eq. (4.32), in order to have $\hat{\sigma}_i(n) = O(\varepsilon)$ we require $n = O\left(\max\{\frac{d^2\sigma_i^2}{\varepsilon^4}, \frac{L^2}{\varepsilon^2}, dM^2\}\right)$. From the properties of the zero-mean noise, to have $\sigma_i(n) = O(\varepsilon^2)$ we require $n = O(\frac{\sigma_i^2}{\varepsilon^4})$. Thus, we can prove the following corollary of the previously proven Theorems 4 to 6 particularly for the zeroth-order information case:

Corollary 2. *We get the following sample complexities for the zeroth-order information case, using* $\nu = \min\left\{\frac{\alpha_t^i}{2\|\nabla f^i(x_t)\| + \sqrt{\alpha_t^i M_i}}, \frac{\alpha_t^i}{2mM_iD}, \frac{\eta}{2mM_0}\right\}$ *:*

- *For the non-convex problem, LB-SGD returns x_t such that is ε-approximate KKT point after at most $N = O(\frac{d^2\sigma_i^2}{\varepsilon^7})$ measurements with probability $1 - \hat{\delta}$.*
- *For the convex problem, LB-SGD returns x_t such that $\mathbb{P}\{f^0(x_t) - \min_{x\in\mathcal{X}} f^0(x) \leq \varepsilon\} \geq 1 - \hat{\delta}$ after at most $N = \tilde{O}(\frac{d^2\sigma_i^2}{\varepsilon^6})$ measurements.*
- *For the strongly-convex problem, LB-SGD returns x_t such that $\mathbb{P}\{f^0(x_t) - \min_{x\in\mathcal{X}} f^0(x) \leq \varepsilon\} \geq 1 - \hat{\delta}$ after at most $N = \tilde{O}(\frac{d^2\sigma_i^2}{\varepsilon^5})$ measurements.*
- *Moreover, all the query points of LB-SGD are feasible for (P) with probability at least $1 - \hat{\delta}$.*

4.5 Non-smooth optimization

In this section, we extend our results to the non-smooth optimization setting. If the objective and constraint functions f^i are assumed to be only L_i-Lipschitz continuous and not necessarily differentiable, then we cannot define the KKT point directly. However, using the notion of the ν-smoothed function (see Definition 1), we define the ν-smoothed approximation of the problem (P) as follows:

$$\min_{x\in\mathbb{R}^d} f_\nu^0(x) \tag{S}$$
$$\text{s.t. } f_\nu^i(x) \leq 0, \quad i \in [m],$$

in which we denote the smoothed set by $\mathcal{X}_\nu := \{x \in \mathbb{R}^d : f^i_\nu(x) \le 0 \;\; \forall\, i \in [m]\}$. Then, we can analyze the convergence to a stationary point of the ν-smoothed problem. We propose to address the problem (S), in contrast to the initial problem (P) in the smooth case.

Smoothing properties We can show the following properties:

Fact 5. *Let functions f^i are L_i-Lipschitz continuous for all $i \in \{0,\dots,m\}$. Then*

1) we can bound the deviation of the smoothed approximation as follows [HL16]:

$$|f^i(x) - f^i_\nu(x)| \le \nu L_i.$$

2) the gradient $\nabla f^i_\nu(x)$ is M^i_ν-Lipschitz continuous with $M^i_\nu \le \frac{2\sqrt{d}L_i}{\nu}$, i.e., $f^i_\nu(x)$ is M^i_ν-smooth;

3) the function f^i_ν is L_i-Lipschitz continuous.

For the proof of property 1) see [FKM05], of property 2) see Section B.8, and for the proof of property 3) see Section B.9. For the convex case, the property 2) was proved by Yousefian, Nedić, and Shanbhag [YNS10].

Gradient estimator In the non-smooth case, we can still estimate the gradients of the smoothed functions f^i_ν using the randomized sampling procedure. Here the functions f^i_ν are M^i_ν-smooth with $M^i_\nu > 0$ which we specify later. If we assume the estimation procedure defined in Algorithm 5, we have access to unbiased stochastic estimated gradients $G^i_{\nu,n}(x_t)$ and *biased* zeroth-order measurements $f^i_\nu(x_t)$, since $F^i(x_t, \xi^i_t)$ are centered at $f^i(x_t)$. But we can slightly modify an estimator, in order to have an unbiased estimators of both objective and constraints.

Algorithm 6 Zeroth-order gradient-value estimator $F^i_n(x,\xi), G^i_{\nu,n}(x,\xi)$

```
1: Input: $F^i(\cdot,\xi), i \in \{0,\dots,m\}$, $x \in D$, $\nu > 0$, $n \in \mathbb{N}$;
2: Sample $n$ directions $b_j \sim \mathcal{U}(\mathcal{B}^d)$, $s_j \sim \mathcal{U}(\mathcal{S}^d)$, sample $F^i(x+\nu b_j, \xi^{i-}_j)$, sample $F^i(x+\nu s_j, \xi^{i+}_j)$, and $F^i(x, \xi_j)$ ;
3: Output:
```

$$F^i_{\nu,n}(x,\xi) := \frac{\sum_{j=1}^n F^i(x+\nu b_j, \xi_j)}{n} \tag{4.33}$$

$$G^i_{\nu,n}(x,\xi) := \frac{d}{n}\sum_{j=1}^n \frac{F^i(x+\nu s_j, \xi^{i+}_j) - F^i(x, \xi^{i-}_j)}{\nu} s_j, \tag{4.34}$$

The above estimator requires 1.5 times more samples since it requires an additional averaging over the measurements on the ball. These estimators defined in Algorithm 6 have the following properties:

Lemma 11. *Let $f^i : \mathbb{R}^d \to \mathbb{R}$ be L_i-Lipschitz continuous. Let f^i_ν be its ν-smoothed approximation. Let $F^i_{\nu,n}(x, \xi)$ and $G^i_{\nu,n}(x, \xi)$ are defined according to the Algorithm 6, then*

1) Both estimators are unbiased, i.e., $\mathbb{E}F^i_{\nu,n}(x, \xi) = f^i_\nu(x)$ and $\mathbb{E}G^i_{\nu,n}(x, \xi) = \nabla f^i_\nu(x)$, that is $\hat{b}_i = 0$;

2) The variance of $F^i_{\nu,n}(x, \xi)$ is bounded with

$$\sigma_i^2(n) = \frac{4L_i^2\nu^2}{n} + \frac{\sigma_i^2}{n}. \tag{4.35}$$

3) The variance of $G^i_{\nu,n}(x, \xi)$ is bounded as follows:

$$\hat{\sigma}^2(n) = \frac{(d+1)^2}{n}\left(L_i^2 + \frac{2\sigma_i^2}{\nu^2}\right); \tag{4.36}$$

For the proof see Appendix B.10.

Adaptive step-size In this case we can bound the smoothness of the log barrier as follows:

$$M_2(x_t) := M^0_\nu + \eta\sum_{i=1}^m \frac{M^i_\nu}{\alpha^i_t} + 4\eta\sum_{i=1}^m \frac{L_i^2}{(\alpha^i_t)^2} = \frac{\sqrt{d}L_0}{\nu} + \eta\sum_{i=1}^m \frac{\sqrt{d}L_i}{\nu\alpha^i_t} + 4\eta\sum_{i=1}^m \frac{L_i^2}{(\alpha^i_t)^2}, \tag{4.37}$$

The estimator $\bar{\alpha}^i_t(\nu) = F^i_{n,\nu}(x)$, $\bar{\alpha}^i_t(\nu) = F^i_{n,\nu}(x)$. The lower bound on α^i_t should work on both smoothed and the original (for safety) constraint functions, therefore we can set $\underline{\alpha}^i_t = \min\{\bar{\alpha}^i_t(\nu) - \frac{\sigma_i + 2L_i\nu}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}, \bar{\alpha}^i_t - \frac{\sigma_i}{\sqrt{n}}\sqrt{\log\frac{1}{\delta}}\}$, for which we have

$$\mathbb{P}\{\underline{\alpha}^i_t \le \min\{-f^i_\nu(x_t), -f^i(x_t)\}\} \ge 1 - \delta.$$

In the non-smooth case there is no much sense of bounding the adaptive γ_t using smoothness, therefore bound adaptive γ_t directly using the L_i-Lipschitz continuity of each constraint i, as shown in Figure 4.3. In particular, we use the following statement:

Lemma 12. *Given that the point x_t is such that $\max\{f^i(x_t), f^i_\nu(x_t)\} < 0$, the next point x_{t+1} generated by the LB-SGD algorithm with $\gamma_t \le \frac{\underline{\alpha}^i_t}{2L_i\|g_t\|}$ with high probability satisfies: $\mathbb{P}\{f^i_\nu(x_{t+1}) \le \frac{1}{2}f^i_\nu(x_t)\} \ge 1 - \delta$ and $\mathbb{P}\{f^i(x_{t+1}) \le \frac{1}{2}f^i(x_t)\} \ge 1 - \delta$.*

Proof. First, note that the condition on γ_t ensures that $x_{t+1} \in \mathbb{R}^d$ lies inside the ball around x_t with radius $\frac{\min\{-f^i_\nu(x_t), -f^i_\nu(x_t)\}}{2L_i}$ with probability $1 - \delta$, i.e.,

$$\mathbb{P}\left\{\|x_{t+1} - x_t\| \le \frac{-f^i_\nu(x_t)}{2L_i}\right\} \ge 1 - \delta. \tag{4.38}$$

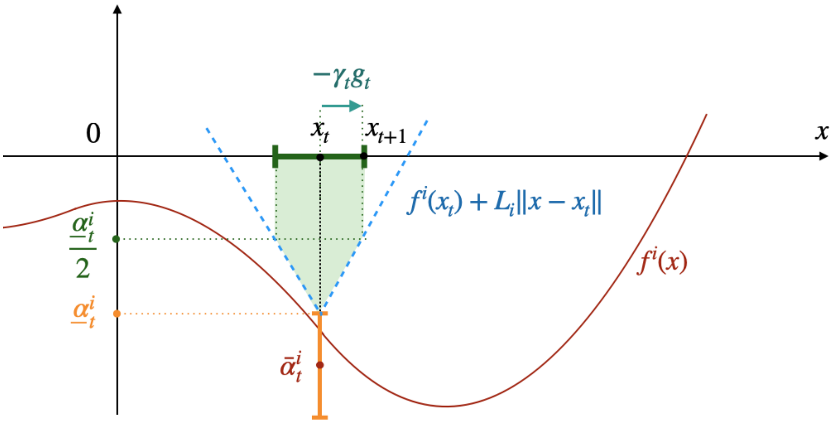


Figure 4.3: Illustration of the adaptivity. Step size γ_t is chosen such that the liner upper bound based on Lipschitz continuity (*blue*) on the constraint guarantees $f^i(x_{t+1}) \leq f^i(x_t)/2$. $\underline{\alpha}_t^i$ is the lower bound on $\alpha_t^i = -f^i(x_t)$, constructed based on the mean estimator $\bar{\alpha}_t^i$. By the *orange* interval we denote the confidence interval for α_t^i. By the *green* interval we denote the adaptive region for x_{t+1} based on the requirement $f^i(x_{t+1}) \leq f^i(x_t)/2$.

Then, we have $\forall i \in [m]$ $f_\nu^i(x_{t+1}) \overset{\textcircled{1}}{\leq} f_\nu^i(x_t) + L_i\|x_{t+1} - x_t\| \overset{\textcircled{2}}{\leq} f_\nu^i(x_t) + L_i\frac{-f_\nu^i(x_t)}{2L_i} = \frac{f_\nu^i(x_t)}{2}$, with probability $1 - \delta$, where the first inequality $\textcircled{1}$ is due to the L_i-Lipschitz continuity (see Fact 5), and the second one $\textcircled{2}$ is due to Eq. (A.12). The same statement holds for $f^i(x_t)$. ■

Then, we define the step-size to be

$$\gamma_t = \min\left\{\min_{i\in[m]}\left\{\frac{\underline{\alpha}_t^i}{2L_i}\right\}\frac{1}{\|g_t\|}, \frac{1}{\hat{M}_2(x_t)}\right\}. \tag{4.39}$$

Changing the step-size influences its lower bound (see Lemma 13), in particular, it changes the constant C that enters all the bounds in Theorems 4 to 6. We can lower bound the corresponding γ_t as follows:

Lemma 13. *If $\underline{\alpha}_t^i \geq c\eta$ for $c > 0$, then we have $\mathbb{P}\{\gamma_t \geq C\eta\} \geq 1 - \delta$ with C defined by*

$$C := \frac{c}{L^2}\min\left\{\frac{1}{2(1+\frac{m}{c})}, \frac{1}{\sqrt{d}(1+\frac{m}{c}) + \frac{4}{c}}\right\} \tag{4.40}$$

Proof. Indeed, using $\alpha_t^i \geq c\eta$ we get $\|g_t\| \leq L(1+\frac{m}{c})$ and $M_2(x) \leq \frac{\sqrt{d}L}{\nu}(1+\frac{m}{c}) + 4\frac{L^2}{c^2\eta}$. Using $\nu = \frac{\eta}{L}$. Therefore,

$$\gamma_t = \min\left\{\min_{i\in[m]}\left\{\frac{\underline{\alpha}_t^i}{2L_i}\right\}\frac{1}{\|g_t\|}, \frac{1}{\hat{M}_2(x_t)}\right\} \geq \min\left\{\frac{c\eta}{2L}\frac{1}{L(1+\frac{m}{c})}, \frac{1}{\frac{\sqrt{d}L}{\nu}(1+\frac{m}{c}) + 4\frac{L^2}{c^2\eta}}\right\}$$

Here, if we keep smoothing factor proportional to η, e.g., $\nu = \min\{\eta, \frac{\underline{\alpha}_t^i}{L_i}\}$ then we 1) satisfy safety requirements, 2) guarantee $\nu \geq \eta \min\{1, \frac{c}{L}\} = \Omega(\eta)$. Then

$$\gamma_t \geq \min\left\{\frac{c\eta}{2L^2(1+\frac{m}{c})}, \frac{c\eta}{\sqrt{d}L^2(1+\frac{m}{c})+4\frac{L^2}{c}}\right\} = \frac{c\eta}{L^2}\min\left\{\frac{1}{2(1+\frac{m}{c})}, \frac{1}{\sqrt{d}(1+\frac{m}{c})+\frac{4}{c}}\right\}$$

■

Note that the constant C now depends on d as $\Omega(\frac{1}{\sqrt{d}})$. The keeping distance property showing $\alpha_t^i \geq c\eta$ is independent on smoothness, and one can check that its proof still holds with the same constants. Therefore, only constant C changes, that influences only on the constants in the convergence rates. The rest convergence results hold the same.

Setting the sampling radius ν and bounding the sample complexity In order to have the deviation from the original problem of an order η, we again require $\nu = O(\eta)$, see Fact 5. Moreover, in order to guarantee safety of all the measurements in the radius of ν around the current point x_t we require $\nu \leq \frac{\alpha_t^i}{L} = O(\eta)$. Hence, from the above Lemma 10, the variance of the estimated gradient with $\nu = O(\eta)$ is $\hat{\sigma}_i^2(n) = \frac{1}{n}O\left(\max\{\frac{d^2\sigma_i^2}{\eta^2}, L_i^2\}\right)$. In order to have $\hat{\sigma}_i(n) = O(\eta)$ we require $n = O\left(\max\{\frac{d^2\sigma_i^2}{\eta^4}, \frac{L^2}{\eta^2}\}\right)$. In order to have $\sigma_i(n) = O(\eta^2)$ we require $n = O(\max\{\frac{\sigma_i^2}{\eta^4}, \frac{L_i^2\nu^2}{\eta^4}\}) = O(\max\{\frac{\sigma_i^2}{\eta^4}, \frac{L^2}{\eta^2}\})$. Recall that for the final round $\varepsilon = O(\eta)$.

Thus, we can observe the following corollary of the previously proven Theorems 4 to 6 particularly for the zeroth-order information case:

Corollary 3. *We get the following sample complexities for the zeroth-order information case, using $\nu = \min\{\eta, \frac{\alpha_t^i}{L}\}$ and using the fact that the constant $C = \Omega(\frac{c^2}{\sqrt{d}L^2})$ is dependent on d now:*

- *For the non-convex problem, LB-SGD returns x_t such that is ε-approximate KKT point of (S) after at most $T = O(\frac{1}{C\eta^3}) = O(\frac{\sqrt{d}}{\eta^3})$, and $N = O(\frac{d^{2.5}\sigma_i^2}{\varepsilon^7})$ measurements.*
- *For the convex problem, LB-SGD returns x_t such that $\mathbb{P}\{f_\nu^0(x_t) - \min_{x\in\mathcal{X}_\nu} f_\nu^0(x) \leq \varepsilon\} \geq 1-\delta$ after at most $T = O(\frac{\sqrt{d}}{\eta^2})$, and $N = O(\frac{d^{2.5}\sigma_i^2}{\varepsilon^6})$ measurements.*
- *For the strongly-convex problem, similarly, LB-SGD returns x_t such that $\mathbb{P}\{f_\nu^0(x_t) - \min_{x\in\mathcal{X}_\nu} f_\nu^0(x) \leq \varepsilon\} \geq 1-\delta$ after at most $N = O(\frac{d^{2.5}\sigma_i^2}{\varepsilon^5})$ measurements.*
- *Moreover, all the query points of LB-SGD are feasible for (P) and (S) with probability at least $1-\delta$.*

4.6 Simulations

In this section, we demonstrate the empirical performance of our method when optimizing smooth synthetic functions on simulations and compare it to other existing non-linear safe learning approaches. All the experiments in this subsection were carried out on a Mac Book Pro 13 with 2.3 GHz Quad-Core Intel Core i5 CPU and with 8 GB RAM. The code corresponding to the experiments in this subsection can be found under the following link: https://github.com/Ilnura/LB_SGD.

Numerical stability. First, we note that to improve numerical stability, we slightly modify the steps of our method for practical applications. Recall that the log barrier gradient estimator is $g_t \leftarrow G_n^0(x_t,\xi_t)+\eta\sum_{i=1}^m \frac{G_n^i(x_t,\xi_t)}{-F_n^i(x_t,\xi_t)}$. Due to noise, the value of $-F_n^i(x_t,\xi_t)$ might become infinitely close to zero or negative, which leads to g_t blowing up or being unreliable. Therefore, we denote by $\bar{\alpha}_t^i$ the truncated value measurements $-F_n^i(x_t,\xi_t)$ with small truncation parameter $a > 0$, that is $\bar{\alpha}_t^i = [-F_n^i(x_t,\xi_t)]_a := \begin{cases} -F_n^i(x_t,\xi_t) & , -F_n^i(x_t,\xi_t) > a \\ a & , -F_n^i(x_t,\xi_t) \leq a \end{cases}$. Based on the above, we use the following estimator for the first-order stochastic optimization at point x_t: $g_t = G_n^0(x_t,\xi_t)+\eta\sum_{i=1}^m \frac{G_n^i(x_t,\xi_t)}{\bar{\alpha}_t^i}$.

Convex objective and constraints

We first compare our safe method LB-SGD with SafeOpt [Sui+15b; BKS16] and LineBO [Kir+19], on a simple synthetic example.

We consider the quadratic problem with linear constraints $\min_{x\in\mathbb{R}^d} \|x-x_0\|^2/4d$, s.t. $Ax \leq b$, where $x_0 = [2,\dots,2]$ and $A = \begin{bmatrix} I_d \\ -I_d \end{bmatrix}$, $b = \mathbf{1}/\sqrt{d}$. The optimum of this problem is on the boundary. We assume that the linearity of the constraints is unknown, hence for SafeOpt we use the Gaussian kernel. For dimensions $d = 2,3,4$ we carry out the simulations with standard deviation $\sigma = 0.001$ of an additive noise Figure 4.4 averaged over 10 different experiments. For $d = 2$ we run SafeOpt, and for $d = 3,4$ we run SafeOptSwarm, which is a heuristic making SafeOpt updates more tractable for slightly higher dimensions [BKS16]. For SafeOpt and LineBO methods, instead of plotting the accuracy and constraints corresponding to x_t, we plot the smallest accuracy and biggest constraint seen up to the step t (for sake of interpretability of the plots). Even for $d = 4$, LB-SGD is already notably more sample efficient compared to both SafeOpt and LineBO. Moreover, LB-SGD significantly outperforms SafeOpt over computational cost and memory usage. It is well known that SafeOpt's sample complexity and computational cost can exponentially depend on the dimensionality. In contrast, the complexity of LB-SGD depends on d polynomially. The runtimes of the above experiments, in seconds, are shown in Table 4.1 and Figure 4.6.

d	2	3	4
SafeOpt (SafeOptSwarm)	4.289	114.406	212.514
LineBO	8.180	17.837	40.8
LB-SGD	0.429	0.895	0.781

Table 4.1: Average runtime dependence on dimensionality d (in seconds). Importantly, the wall-clock time of LB-SGD remains roughly constant as we increase d, noting that we only increase the number of measurements per iteration, but not the number of iterations.

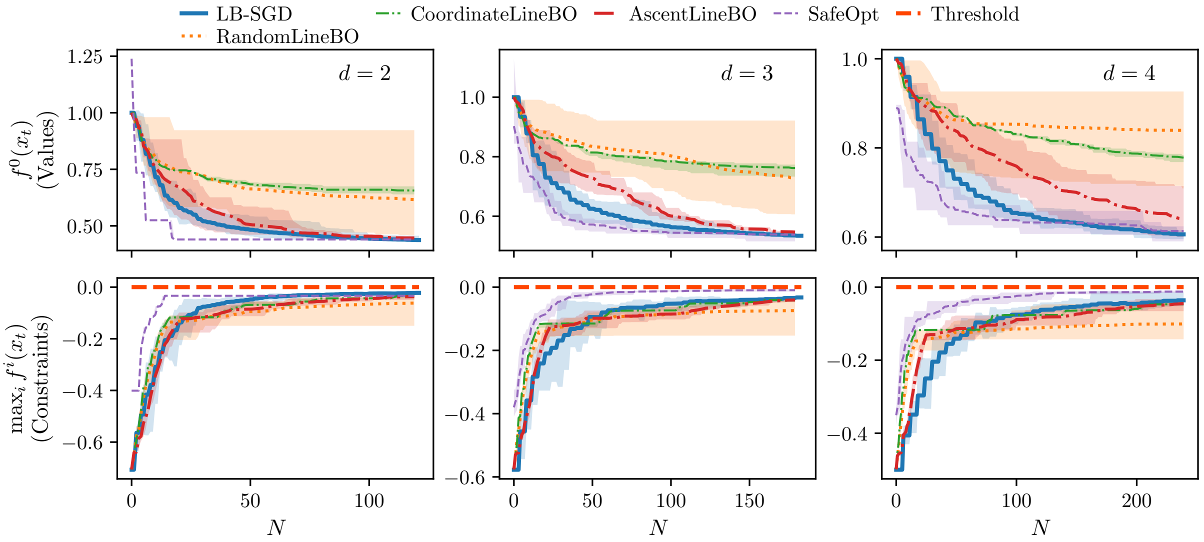


Figure 4.4: Accuracy (upper plots) and constraints (lower plots) of LB-SGD and SafeOpt for $d = 2, 3, 4$, averaged over 10 samples. t here is the amount of zeroth-order oracle calls. In these experiments, for LB-SGD we decrease $\eta_{k+1} = 0.85\eta_k$ gradually every $T_k = 3$ steps with $n_k = [\frac{d}{2}]$ value measurements at each step. Already for $d = 4$ LB-SGD starts outperforming all BO-based methods on this problem in terms of the sample complexity.

d	2	3	4
SafeOpt (SafeOptSwarm)	5.308	44.909	63.019
LineBO	7.584	10.593	13.293
LB-SGD	0.294	0.332	0.324

Table 4.2: Run-time (in seconds) as dependent on dimensionality d (Rosenbrock benchmark).

Non-convex objective and constraints

As a non-convex example, we consider the Rosenbrock function, a common benchmark for black-box optimization, with quadratic constraints. In particular, we consider the following problem

$$\min_{x\in\mathbb{R}^d} \sum_{i=1}^{d-1} 100\|x_i - x_{i+1}\|^2 - \|1 - x_i\|^2,$$
$$\text{s.t. } \|x\|^2 \leq r_1^2,\ \|x - \hat{x}\|^2 \leq r_2^2.$$

We set $r_1 = 0.1$, $r_2 = 0.2$, $\hat{x} = [-0.05, \ldots, -0.05]$. The optimum of this problem is on the boundary of the constraint set. We show the comparison of LB-SGD and SafeOpt on Figure 4.5. Again, for $d = 2$ we run SafeOpt, and for $d = 3, 4$ we run SafeOptSwarm. Here, on the constraints plot of SafeOpt and LineBO we again plot the highest value of the constraints over all points explored so far.

The run-times of LineBO and SafeOpt are demonstrated in Table 4.2 and Figure 4.6.

Note that the second problem is easier for BO methods than the first one. It is related

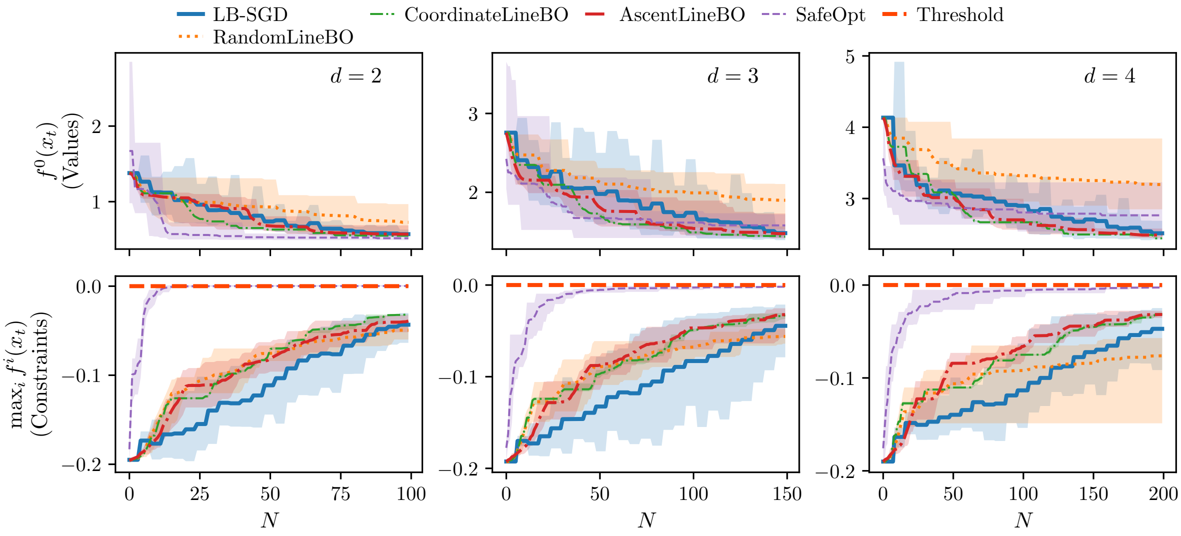


Figure 4.5: Accuracy and constraints of LB-SGD and SafeOpt for $d = 2, 3, 4$, averaged over 10 samples. t here is the amount of zeroth-order oracle calls. In these experiments, for LB-SGD we decrease $\eta_{k+1} = 0.7\eta_k$ gradually every $T_k = 5$ steps with $n_k = d - 1$ value measurements at each step. On this problem, for $d = 4$ we observe that LB-SGD performs better that SafeOpt, and comparable to LineBO.

to the fact that in the first problem, the number of constraints (and therefore, the number of GPs) is higher and grows with dimensionality ($m = 4, 6, 8$). In contrast, there are always only two constraints for the second problem. As one can see, our approach is significantly cheaper in computational time than SafeOpt. This is, of course, at the price of finding only a local minimum, not the global one.

Comparison with LineBO in higher dimensions

In higher dimensions, it is well known that SafeOpt is not tractable. Therefore, we compare our method only with LineBO [Kir+19]. This method scales significantly better with dimensionality than the classical BO approaches. The method was demonstrated to be efficient in the unconstrained case and in cases where the solution lies in the interior of the constraint set. The authors proved the theoretical convergence in the unconstrained case and the safety of the iterations in the constrained case. However, in contrast to our method, this approach has a drawback that we discuss below. At each iteration, LineBO samples a direction (at random, an ascent direction of the objective, or a coordinate direction). Then it solves a *1-dimensional* constrained optimization along this direction, using SafeOpt. After optimizing along this direction, it samples another direction starting from the current point. The drawback of this approach is that when the solution is on the boundary, LineBO might get stuck on the wrong point on the boundary. In such a case, it might be difficult for it to find a safe direction of improvement too close to the boundary. See Figure 4.7 for the illustration of this potential problem. Furthermore, the higher dimension, the harder it is to sample a suitable direction. We demonstrate that

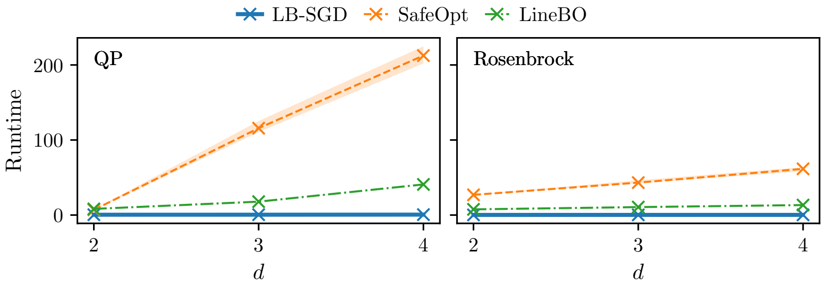


Figure 4.6: Run-times of LB-SGD and SafeOpt for $d = 2, 3, 4$, averaged over 10 samples, in seconds. t here is the amount of zeroth-order oracle calls. We can observe that LB-SGD is a significantly cheaper approach in terms of the computational cost compared to both BO-based methods with growing dimensions.

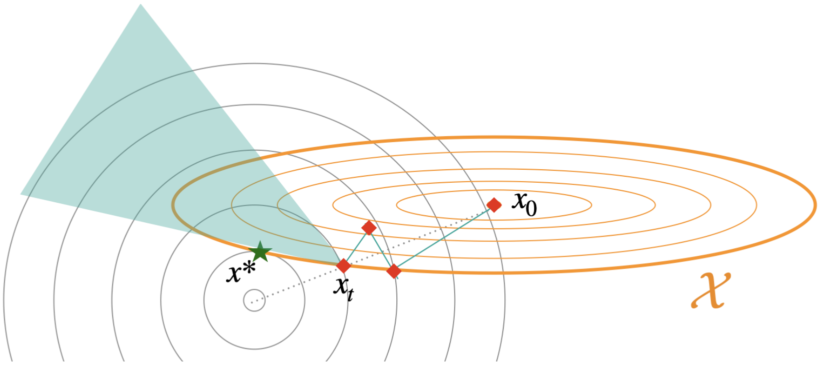


Figure 4.7: Illustration of the LineBO behavior. At point x_t not every direction allows the safe improvement (only the directions lying in the green sector). Therefore, the LineBO method might get stuck sampling the wrong directions. On this example, the closer to the solution, the narrower is the improvement sector.

empirically in application to the following problem:

$$\min_{x \in \mathbb{R}^d} -\exp^{-4\|x\|^2}, \tag{4.41}$$

$$\text{s.t. } \langle x - \hat{x}, A(x - \hat{x}) \rangle \leq r^2, \tag{4.42}$$

with $r = 0.5$ and $A = \operatorname{diag}(3, 1.2, \dots, 1.2)$. On Figure 4.8 we demonstrate the comparison of LineBO and LB-SGD methods on the above problem for dimensionalities $d = 2, 10, 20$. We report the run-times in Table 4.3.

To compare, in the case when the solution is in the interior of the constraint set achieved by setting $r = 10$ (that is, if the constraints do not influence the solution), the LineBO approach does not have this issue and can still be very efficient (see Figure 4.9).

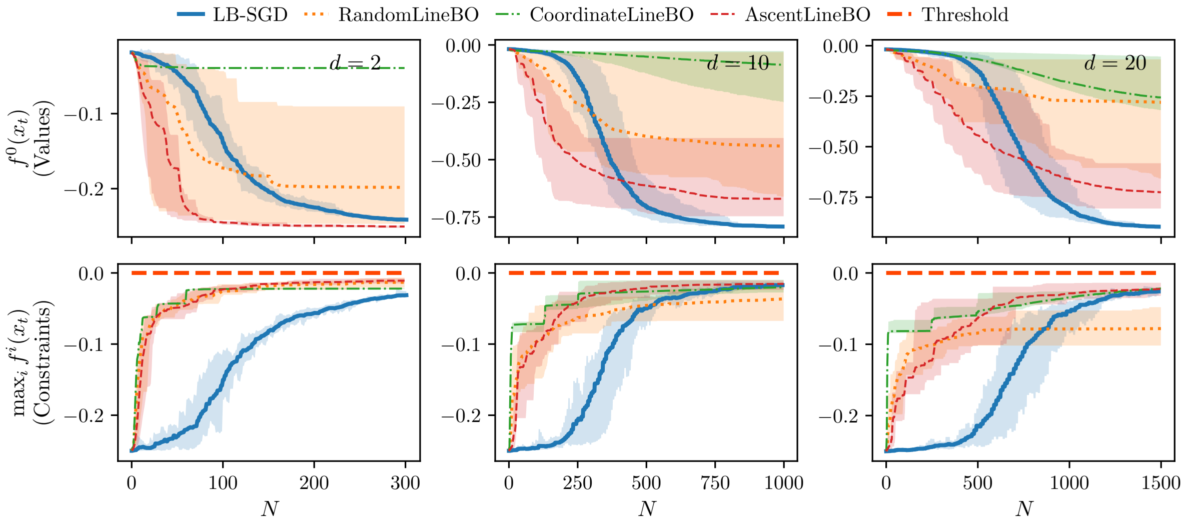


Figure 4.8: Accuracy and constraints of LB-SGD and SafeRandomLineBO for $d = 2, 10, 20$, averaged over 10 samples. t here is the amount of zeroth-order oracle calls. In these experiments, for LB-SGD we decrease $\eta_{k+1} = 0.85\eta_k$ gradually every $T_k = 3$ steps with $n_k = [\frac{d+1}{2}]$ value measurements at each step.

d	2	10	20
LB-SGD	0.828	2.186	2.676
LineBO	12.883	298.097	1038.459

Table 4.3: Runtime (in seconds) dependence on dimensionality d on the negative Gaussian minimization benchmark. We can observe that LineBO is significantly more expensive in computational cost (for the same number of queried points).

4.7 Conclusion

In this chapter, we addressed the problem of sample and computationally efficient safe learning. We proposed an approach based on logarithmic barriers, which we optimize using SGD with adaptive step sizes. We analytically proved its safety during the learning and analyzed the convergence rates for non-convex, convex, and strongly-convex problems. We empirically demonstrated the performance of our method in comparison with other existing methods. We showed that 1) its sample and computational complexity scale efficiently to high dimensions, and; 2) it keeps optimization iterates within the feasible set with high probability. Additionally, we demonstrated the efficiency of the log barrier approach for high-dimensional constrained reinforcement learning problems.

While not requiring to explicitly specify a prior (in the Bayesian sense, as considered in safe Bayesian optimization), our method does involve hyper-parameters such as η_0, η-decrease rate parameter ω, amount of steps per episode T_k, and exhibits sensitivity to the noise. Also, in the non-convex case, it can converge only to a local minimum, as any other descent optimization approach. However, it is easy to implement and has efficient computational performance due to cheap updates. Therefore, LB-SGD is better suited to

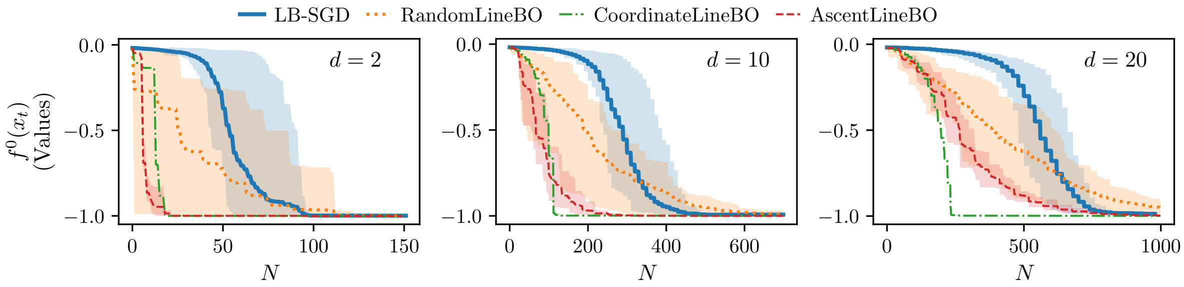


Figure 4.9: Accuracy and constraints of LB-SGD and SafeRandomLineBO for $d = 2, 10, 20$, averaged over 10 samples. t here is the amount of zeroth-order oracle calls. In these experiments, for LB-SGD we decrease $\eta_{k+1} = 0.85\eta_k$ gradually every $T_k = 3$ steps with $n_k = [\frac{d+1}{2}]$ value measurements at each step.

problems of high scale.

For future work, it would be exciting to take the best of both worlds and combine the BO approaches that allow us to build and use a global model with our simple and cheap safe descent approach based on log barriers.

CHAPTER 5

Applications

In this chapter, we demonstrate the particular applications in robotics, manufacturing, and reinforcement learning, where safety during the learning is required. The code corresponding to the experiments described in the Section 5.1 and Section 5.2 can be found under this link: `https://github.com/Ilnura/Thesis_applications`.

5.1 Turning process optimization

Let us start with the simple problem of parameters tuning of the cutting machine.

We consider the scenario of a cutting machine [Mai+18] which has to produce certain tools and optimize the cost of production by tuning the turning process parameters such as the feed rate f and the cutting speed ν_c. For the turning process we need to minimize a non-convex cost function $C(x)$, where the decision variable is $x = (\nu_c, f) \in \mathbb{R}^2$. The constraints include box constraints and a non-convex quality roughness constraint $R(x)$. We perform realistic simulations, by using the cost function and constraints estimated from hardware experiments with artificially added normally distributed noise $\xi \sim N(0, \sigma^2)$. The obtained non-convex smooth optimization problem with concave objective and convex constraints is:

$$\begin{aligned}
\min_{x\in\mathbb{R}^2} \quad & C(x) = t_c(x)\left(C_M + \frac{C_I}{T(x)}\right)\\
\text{subject to} \quad & R(x) \le 0.7, x_1 = \nu_c \in [100, 200],\\
& \qquad\qquad\quad x_2 = f \in [0.08, 0.16].
\end{aligned}$$

Here the values are

$$\begin{aligned}
& t_c(x) = \frac{LD\pi}{\nu_c f},\\
& T(x) = 127.5365 - 0.84629\nu_c - 144.21f + 0.001703\nu_c^2 + 0.3656\nu_c f,\\
& R(x) = 0.7844 - 0.010035\nu_c + 7.0877f + 0.000034\nu_c^2 - 0.018969\nu_c f,\\
& C_I = 40,\ C_M = 50.
\end{aligned}$$

Note that we assume the box constraints to be known, i.e., not corrupted with noise. However, the roughness constraint $R(x)$ and the cost $C(x)$ are assumed to be unknown and we only can measure their noisy values. Hence, this problem is an instance of the safe learning problem formulated in (P). In this case, we set $f^0(x) = C(x)$, $f^1(x) = R(x) - 0.7$, and $f^2, \dots, f^5$ are determined by the box constraints. More details are presented by Maier, Rupenyan, Zwicker, Akbari, and Wegener [Mai+18], who proposed to use Bayesian optimization with inequality constraints [Gar+14] to solve the problem. Although the Bayesian optimization used there indeed requires a small number of measurements, it is not safe and hence may require several measurements to be taken in the unsafe region. The roughness constraints are not fulfilled for unsafe measurements, i.e., the tools produced during unsafe experiments could not be realised in the market. That is why safety is necessary for this problem. Although there exist safe Bayesian optimization [Sui+15b; Ber+17] methods, they also require strong prior knowledge in terms of suitable kernel function. For this problem, Assumptions 1 to 3 hold, and the satisfaction of Assumption 4 (MFCQ) can be observed from the plot of the feasibility set on Figure 5.1 (denoted by the green colour) for such a small problem. We solve barrier sub-problem iteratively using LB-SGD with decreasing $\eta_{k+1} = \omega\eta_k$, where we fix $\omega = 0.7$. We set $\sigma = 0.001, L = 2, M = 30, \delta = 0.01$ and re-scaled $\nu_c' = 0.001\nu_c$ so that $\nu_c' \in [0.1, 0.2]$. In Figure 5.1 we compare the performance of SafeOpt [BKS16] with RBF kernel and the performance of of LB-SGD starting from the point $x_0 = (\nu_c', f) = (0.18, 0.11)$ with decreasing $\eta_{k+1} = 7\eta_k$ every $T_k = 7$ iterations, and $\eta_0 = 0.1$. We average the convergence and constraints plots over 10 independent runs. We only made two measurements at each iteration. If σ is bigger, we need to make $n_t = (\sigma/\eta)^2$ measurements per iteration, otherwise the best accuracy it can achieve is proportional to σ. In all the runs the LB-SGD method converges to an optimum and the constraints are not violated. The initial feasible points we choose manually, in this small example it is not a hard task.

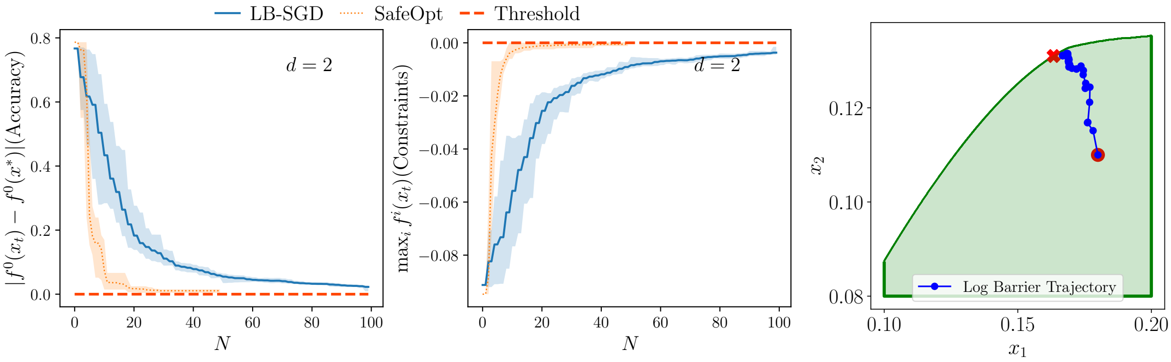


Figure 5.1: Convergence of the cost (*left*), maximal constraint plot (*middle*) averaged over 10 runs of LB-SGD and SafeOpt, and optimization trajectory of LB-SGD (*right*). In the simulations we considered noise $\xi \sim \mathcal{N}(0, \sigma^2)$ added to the measurements . By the red cross we denote the optimal point of the problem. Blue points denote the optimization trajectory, red point denotes the starting point, green set is the feasibility set.

In Figure 5.2 we demonstrate the convergence from another starting point $x_0 =$

[0.12, 0.09] and another $\eta_0 = 0.3$, and observe that if the starting point is further away from the optimum, and closer to the boundary, it takes longer for LB-SGD to converge as expected from the theory, whereas for the SafeOpt approach it takes almost the same 20 iterations to converge in both cases, i.e., it does not depend on how close the initial point is to the solution.

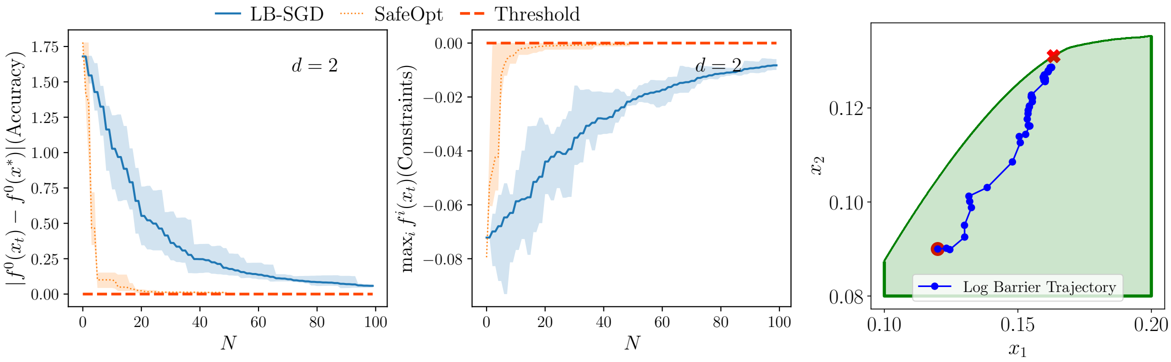


Figure 5.2: Convergence of the cost (*left*), maximal constraint plot (*middle*) averaged over 10 runs of LB-SGD and SafeOpt, and optimization trajectory of LB-SGD (*right*). The starting point $x_0 = [0.12, 0.09]$ is closer to the boundary and further form the solution. We have $N = 100$ measurements, and $T = 50$ iterations in total with $n = 1$ (two measurements per each iteration).

This showcase demonstrates that despite our method on such a small dimensional problem is slower than Bayesian approaches, it does not require knowledge of the kernel, has significantly faster updates, and still is comparable to Bayesian approaches in terms of the sample complexity, especially if the starting point is comparably close to the solution. As we have seen from the higher dimensional examples in the previous chapter, on higher dimensions LB-SGD outperforms SafeOpt. That brings us to the idea that our method could work the best in combination with global BO methods, to take the best from both worlds. For example, one could run the safe Bayesian method on the first few iterations, and next to fine tune the approximate solution using safe local descent method such as LB-SGD. We consider to address this idea in future.

5.2 Learning the controller with log barriers

In this section we demonstrate performance of our approach in application to two control problems formulated via linear controller regulation (LQR).

5.2.1 Convex LQR

Here we demonstrate the performance of our LB-SGD algorithm on the simple finite horizon LQR example with linear dynamics. We select this example since it is a non-trivial convex problem that requires safety. Consider the linear dynamical system: $q_{\tau+1} = Aq_\tau + Bu_\tau$ and the following LQR problem for reaching the target q_{target}:

$$\min_{\mathbf{u}} \ \frac{1}{H}\sum_{\tau=1}^{H} \|q_\tau - q_{\text{target}}\|^2 \tag{5.1}$$

$$\text{s.t.} \ \ q_0 = q^{(0)}, \ q_{\tau+1} = Aq_\tau + Bu_\tau, \tag{5.2}$$

$$q_\tau \in \mathcal{Q}, \qquad \forall \tau = [H] \tag{5.3}$$

The matrices A, B are unknown. The goal is to learn the open-loop controller $\mathbf{u} = (u_0, \ldots, u_{H-1})$ minimizing the LQR objective while not violating the hard constraints (5.3) during the learning. In our case study, we consider the time horizon $H = 10$, the dynamic given by $A = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, the safety set we define artificially in such a way that it is convex, non-linear, and simple to illustrate:

$$\mathcal{Q} = \{q \in \mathbb{R}^2 : \|q\|_2 \le 3, \ \begin{bmatrix} 1 & 1 \end{bmatrix} q \le 3, \ \begin{bmatrix} -1 & -1 \end{bmatrix} q \le 3\}.$$

We assume we have access to zeroth-order noisy oracle of each of these three constraints defining $\mathcal{Q}$ for every state q_t in the generated trajectory: *1)* $\|q_t\|_2 - 3 \le 0$; *2)* $\begin{bmatrix} 1 & 1 \end{bmatrix} q_t - 3 \le 0$; *3)* $\begin{bmatrix} -1 & -1 \end{bmatrix} q_t - 3 \le 0$, thus, in total we have $3H$ constraints. The target state we set to be $q_{\text{target}} = [2.5, 0]$, and the initial state to be $q_0 = [-2, 0]$. The input sequence is initialized as $u_\tau = 0$, $\tau = 0, \ldots, H$. Note that for zero input, the trajectory is stationary and therefore safe, since the initial point is an equilibrium of the system. We consider the noise level of $\sigma = 0.0001$. The parameters of the algorithm are tuned manually to $n_k = 6$ $\eta_0 = 0.25$, $T_k = 7$, $\nu = 0.01$, and $\delta = 0.01$. This problem satisfies the Assumptions 1 to 3, and convexity Assumption 5.

Figure 5.3 a) (*left*) shows the performance of 30 independent runs of each algorithm. During the experiments the method never violated the constraints as can be seen from Figure 5.3 a) (*right*). In Figure 5.3b) we show the trajectory corresponding to the solution obtained using the *scipy.optimize.minimize* package with tolerance = 0.001 as if everything was known (using an unsafe optimization method). The problem itself is non-convex, so this solution is an approximate local optimum. In Figure 5.3c), we compare the solution of the LQR problem obtained (*left*) at iterations 5, (*middle*) at iteration 20, and (*right*) The trajectory corresponding to the input sequence learned by LB-SGD .

5.2.2 Non-convex LQR

In this section, we consider the application to safe iterative controller design. Consider the basic unicycle dynamics $\dot{x} = v\cos\theta, \dot{y} = v\sin\theta, \dot{\theta} = \omega$. Here the states $q = [x, y, \theta]$ describe the spatial coordinates x, y and the direction angle θ. The control inputs $u = [v, \omega]$ describe the speed and the angular velocity. We use Euler-discretized model of the unicycle

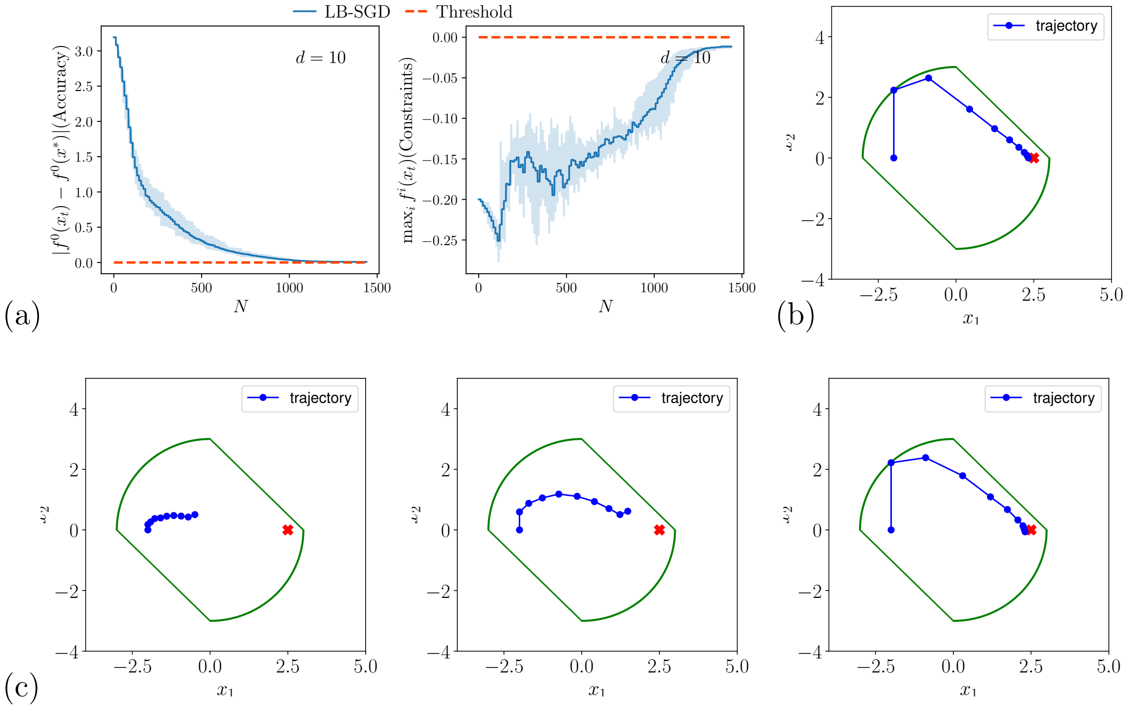


Figure 5.3: a) Illust. of (*left*) convergence and (*right*) maximal constraint value over 30 trials of LB-SGD with $N = 1500$, $n = 6$, $M = 1$, $\sigma = 0.0001$, $\nu = 0.01$, $\eta_0 = 0.1$, with decreasing $\eta_{k+1} = 0.7\eta_k$ every $T_k = 7$ iterations. b) Dynamic trajectory of the optimal $\mathbf{u}^*$ (obtained using *scipy.optimize.minimize* package, 'SLSQP' method, assuming everything is known, infeasible approach during the learning). c) Dynamic trajectory of mid points of LB-SGD at iterations $t = 5$ and $t = 20$ respectively, and the final output $\mathbf{u}_T$. Total number of measurements is $N = 1500$.

dynamics

$$q_{\tau+1} = \begin{bmatrix} x_{\tau+1} \\ y_{\tau+1} \\ \theta_{\tau+1} \end{bmatrix} = \begin{bmatrix} x_\tau + d\tau v_\tau \cos(\theta_\tau) \\ y_\tau + d\tau v_\tau \sin(\theta_\tau) \\ \theta_\tau + d\tau\omega_\tau \end{bmatrix}.$$

We choose a open-loop feedback $\mathbf{u} = [u_0, \dots, u_H] \in \mathbb{R}^{H \times p}$ as the optimizing parameter, where H is the planning horizon. The let sequence determined by $\mathbf{u}$ be denoted by $q_\tau(\mathbf{u})$, $\tau \in [H]$. The goal is to lead the vehicle from a starting point q_0 to a goal destination q_{target} while avoiding collision with high-probability. The cost function is defined by $\frac{1}{H}\sum_{\tau=1}^{H} \|q_\tau(\mathbf{u}) - q_B\|^2 + \frac{1}{10H}\sum_{\tau=1}^{H} \|u_\tau\|^2$. The constraints are formulated such that the trajectory does not collide with the the ball shaped obstacle placed at $[0, 0]^T$ with radius 1. The resulting constrained optimization problem is as follows:

$$\min_{\mathbf{u} \in \mathbb{R}^{3\times 2}} \frac{1}{H}\sum_{\tau=1}^{H} \|q_\tau(\mathbf{u}) - q_{target}\|^2 + \frac{1}{10H}\sum_{\tau=1}^{H} \|u_\tau\|^2$$
$$\text{s.t.} \quad \|(x_\tau(\mathbf{u}), y_\tau(\mathbf{u})) - (x_C, y_C)\|^2 \geq 1, \ \forall \tau \in [H].$$

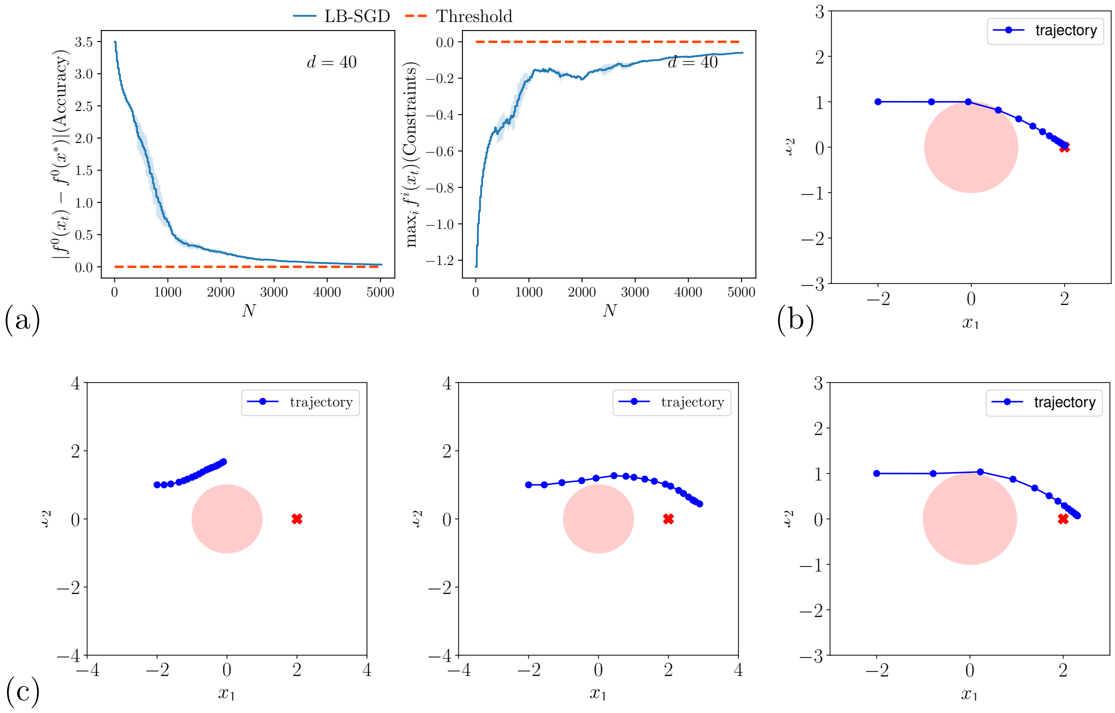


Figure 5.4: a) The objective value (*left*) and maximum constraint value (*right*) for 5 experiments; b) Optimal control trajectory *scipy.optimize.minimize* package, 'COBYLA' unsafe method assuming everything is known); c) Control trajectory obtained by LB-SGD with $\eta_0 = 0.05$, $\omega = 0.93$, $n = 10$, $T_k = 10$, $N = 5000$ at iteration $t = 20$ (*left*), iteration $t = 50$ (*middle*), and the final output at iteration $T = 250$ (*right*). Initial trajectory is the stationary trajectory.

This problem satisfies Assumptions 1 to 3. The bounded diameter is not stated explicitly, but we can find such a ball with diameter D such that the solution with such an additional constraint will not change. In the zeroth-order oracle approach, we assume no knowledge of the dynamics, the constraints or the cost functions. We only assume noisy measurements of the cost function and the constraints. Thus, we address this problem using the LB-SGD algorithm. We set the parameters of the algorithm to $\nu_t = 0.01$ for safety, $M = 40$ set by trial, $n = 10$, $T_k = 10$, $\omega = 0.93$, and initialize the algorithm with a safe stationary control policy such that the agent simply is not moving. The algorithm iteratively improves the controller while avoiding the constraints. The total number of measurements is $N_T = 5000$. In Figure 5.4 a) we demonstrate the achieved results of 20 trials of the stochastic LB-SGD algorithm with the fixed initialization. In none of the trials the constraints were violated. In Figure 5.4 b) we show the trajectory generated by a solution $\mathbf{u}^*$ controller obtained using *scipy.optimize.minimize* package ('COBYLA' unsafe method assuming everything is known). In Figure 5.4 c) we demonstrate an example of the trajectory generated by the controller obtained at iteration 20 (*left*), iteration 50 (*middle*), and the final output at iteration 250 (*right*) during one of the runs of LB-SGD algorithm.

5.3 LB-SGD for safe reinforcement learning

We previously showed LB-SGD performance on a smaller scale, classical black box benchmark problems. In this part, we showcase how LB-SGD scales to more complex, high-dimensional domains arising in RL. This section is based on the experimental part of our paper Usmanova, As, Kamgarpour, and Krause [Usm+22]. It was done in collaboration with Yarden As, who implemented this experiment. The author's contribution was in proposing to use logarithmic barrier steps for the model-based RL and designing the experiment in a suitable way for the safety setup. For our implementation, please see https://github.com/lasgroup/lbsgd-rl.

The safe transfer learning task Recall our introduction example on an autonomous driving car with a well-trained policy π_1 for Zürich. For Lausanne, this policy is still safe but not optimal anymore since Lausanne has different properties. One wants to update this policy to improve its performance in Lausanne by collecting new data using online agent-world interactions (another example is sim-to-real domain adaptation [ZQW20]). At the same time, we would like to keep all the policy updates during the learning process safe and not leading to dangerous situations, which is extremely important for such tasks as autonomous driving in an actual city.

In this section, we demonstrate the performance of our algorithm for safe transfer learning not on real cars, but using the Safety Gym simulator [RAA19]. In particular, we take an algorithm (e.g., LAMBDA [As+22]) that is capable of solving CMDPs in a scalable way but fails in the safe transfer learning task. This method uses the model-based approach, which is a huge advantage for our problem since: (a) model-based approaches are much more sample efficient than the model-free approaches, and (b) the learned model allows to transfer the knowledge not only of the policy pre-trained on the past domain but also the knowledge of the dynamics. Now we show that by changing the optimization algorithm to our algorithm, we can safely transfer between tasks. The main advantage of the log barrier algorithm is that given an initially safe (but sub-optimal) policy, it can remain safe but improve its performance. To demonstrate this, we pre-train a policy on an easy task at the first stage of the experiment. Following that, we switch to a harder task in the second stage. However, we keep safety constraints complexity the same for both stages so that we can transfer safety property from the first to the second stage. Our goal is, given the pre-trained safe policy on a simpler task, to update the policy in a safe way to improve its performance on the harder task. Our algorithm demonstrates the minimum amount of constraint violations during the learning.

5.3.1 Reinforcement learning background

First, let us provide the background required to understand our experiments.

Safety Gym environment We perform our experiment of the Safety Gym simulator [RAA19], based on OpenAI MuJoCo. This is a benchmark suite that treats the safe RL problem with constrained Markov decision processes (CMDP), the dynamics simulator. An agent needs to solve the task while making a smallest amount of obstacles violations.

The benchmark provides several options for choosing the agent such as Point, Car, Doggo, they all have different dynamics. The Safety Gym also provides several options to choose the task, and the complexity of the environment. That is, in options Goal 1 and Goal 2, the task of the agent is to reach the green transparent cylinder target while avoiding several amount of randomly placed obstacles, and Goal 1 has fewer obstacles than Goal 2, Goal 2 is more complicated for safety. In Push 1 and Push 2, the goal is to push the yellow object to a target, and in Button 1 and Button 2 the goal is to press the green button, while avoiding the orange buttons. Safety Gym allows the agent to observe LIDAR simulator, or the images. In our experiments in Safety Gym environment, at each time t the agent gets observations $o_t \in \mathbb{R}^{64\times64\times3}$ which are 64×64 pixels RGB images, taken from the first-point view.

Constrained Markov decision processes The problem of safe reinforcement learning can be viewed as finding a policy that solves a *Constrained Markov decision process* [Alt99]. Briefly, we define a discrete-time episodic CMDP as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mathcal{C})$. At each time step $\tau \in \{0, \dots, T\}$, an agent observes a state $s_\tau \in \mathcal{S}$. Given that state, it decides what action $a_\tau \in \mathcal{A}$ to take next. Then, an unknown transition density $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, $s_{\tau+1} \sim P(\cdot|s_\tau, a_\tau)$ generates a new state. $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function that generates an immediate reward signal observed by the agent. The discount factor $\gamma \in (0, 1]$ weighs the importance of immediate rewards compared to future ones. Lastly, $\mathcal{C} = \left\{c^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \;\middle|\; i \in [m]\right\}$ is a set of cost signals that the agent observes alongside the reward. In our experiments we focus on the case with a single constraint $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which corresponds to hitting any of the obstacles. The goal is to find a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that solves the constrained problem:

$$\max_\pi \underbrace{\mathbb{E}\left[\sum_\tau \gamma^\tau R(s_\tau, a_\tau)\right]}_{J(\pi)} \quad \text{s.t.} \quad \underbrace{\mathbb{E}\left[\sum_\tau \gamma^\tau c(s_\tau, a_\tau)\right] - d^i}_{J_c(\pi)} \leq 0. \tag{5.4}$$

Whereby d is a predefined threshold value for the cost. Note that we take the expectation with respect to all stochasticity induced by the CMDP.

On-policy methods as black-box optimization problems A typical recipe to solve CMDPs at scale is to parameterize the policy with parameters x and use *on-policy* methods. On-policy methods use Monte-Carlo sampling to sample trajectories from the environment, evaluate the policy, and finally update it [Cho+15; Ach+17; RAA19]. By using Monte-Carlo, these methods compute unbiased estimates of the constraints, objective and their gradients, equivalently to the assumptions in Section 2.1. Importantly, the process of sampling trajectories from the CMDP and averaging them to estimate the objective and constraints in Eq. (5.4) is equivalent to querying $f^0(x)$ and $f^i(x), i \in [m]$ as we present in this chapter. However, without deliberately enforcing $x_t \in \mathcal{X}\ \forall t \in \{0, \dots, T\}$, these methods are exposed to use an unsafe policy during learning.

5.3.2 Solving CMDPs with LB-SGD

To demonstrate LB-SGD's ability to keep the policy safe during learning we use LAMBDA [As+22], a model-based algorithm for solving CMDPs. In summary, LAMBDA learns the transition density P from image observations, and uses this learned model to find a (hopefully) optimal policy. The policy is updated with *model-generated, on-policy trajectories*. Clearly, model-generated trajectories are subject to model errors which in turn make the estimation of the objective and constraints biased. As a result, the assumptions in Section 2.1 do not necessarily hold. Nevertheless, this biasedness is subject only to LAMBDA's model inaccuracies, so LB-SGD can still produce safe policies with high utility, as we empirically show in the following section. To employ LB-SGD within LAMBDA, we replace their proposed Augmented Lagrangian [NW06] optimization scheme with LB-SGD.

Model based approach to partially observed states *Internal dynamic model* We consider problems where the agent receives an observation $\mathbf{o}_t \sim P_o(\cdot|\mathbf{s}_t)$ instead of $\mathbf{s}_t$ at each time step, the true state space is hidden. That is, we consider a partially observed Markov Decision Process (POMDP). (In our Safety Gym example the observation is given by the first-person view camera observations.) Then we learn the internal model $P_\theta(a_{\tau-1}, o_\tau)$ which provides the distribution of the latent state s_τ, that is, $s_\tau \sim P_\theta(a_{\tau-1}, o_\tau)$. This model is modeled as the neural network (NN) parametrized by θ (see Appendix C.1). To train it, we use its own loss function $\mathcal{L}_P(\theta, \phi)$ which we define in Appendix C.2 for the interested reader, similarly to Hafner, Lillicrap, Fischer, Villegas, Ha, Lee, and Davidson [Haf+19a].

Critics. Additionally, for the task and safety critics, we use the reward and the cost value functions. We model the reward value function as $v^\pi_\psi(s_t) \approx \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r(s_\tau, a_\tau)|s_t\right]$ which is given as a dense neural network with parameters ψ and discount factor γ. Similarly, we model the cost value by $v^\pi_{\psi_c}(s_t) \approx \left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} c^i(s_\tau, a_\tau)|s_t\right]$ with parameter vector ψ_c. The policy and value models are trained cooperatively as typical in policy iteration: the action model aims to maximize an estimate of the value, while the value model aims to match an estimate of the value that changes as the policy model changes. We use TD(λ) [SB18] to trade-off the bias and variance of the critics with bootstrapping and Monte-Carlo value estimation. We denote $\mathbf{V}_\lambda(s_\tau), \mathbf{V}_{\lambda,c}(s_\tau)$ as the TD(λ) value as in Hafner, Lillicrap, Ba, and Norouzi [Haf+19b], *that is defined recursively using* $v^\pi_\psi(s_\tau)$, $v^\pi_{\psi_c}(s_\tau)$ respectively. Then, the reward value function can be estimated by averaging over the time horizon H as follows $\frac{1}{H}\sum_{t=\tau}^{\tau+H} \mathbf{V}_\lambda(s_t)$, and the cost value function by $\frac{1}{H}\sum_{t=\tau}^{\tau+H} \mathbf{V}_{\lambda,c}(s_t)$, and use them for estimating the stochastic inexact models of $J(\pi_\xi|P), J_c(\pi_\xi|P)$. Here $\mathbf{V}_\lambda(s_t)$ is also dependent on P_θ, π, ψ, and $\mathbf{V}_{\lambda,c}(s_t)$ is also dependent on P_θ, π, ψ_c, we omit these dependencies in notation for simplicity.

Policy. We model the policy as a Gaussian distribution via a neural network with parameters ξ such that $\pi_\xi(a_\tau|s_\tau) = \mathcal{N}(a_\tau; \mathrm{NN}^\mu_\xi(s_\tau), \mathrm{NN}^\sigma_\xi(s_\tau))$. That is, given the state s_τ, the action a_τ is sampled from the normal distribution with the mean returned by $\mathrm{NN}^\mu_\xi(s_\tau)$, and the standard deviation $\mathrm{NN}^\sigma_\xi(s_\tau)$.

The learning loop. In total we can write the loop of collecting the data, and the loop of learning the models as shown at Figure 5.5. At each episode we first run $T = 1000$ agent-environment interactions to collect the data, then at the end of the episode, using the collected data we make 100 gradient steps of the loss: *1)* to update internal model P_θ, *2)* to update the value critics for reward and costs; and *3)* using the updated models, make a safe LB-SGD step to update the policy. The last learning step we describe in the following paragraph.

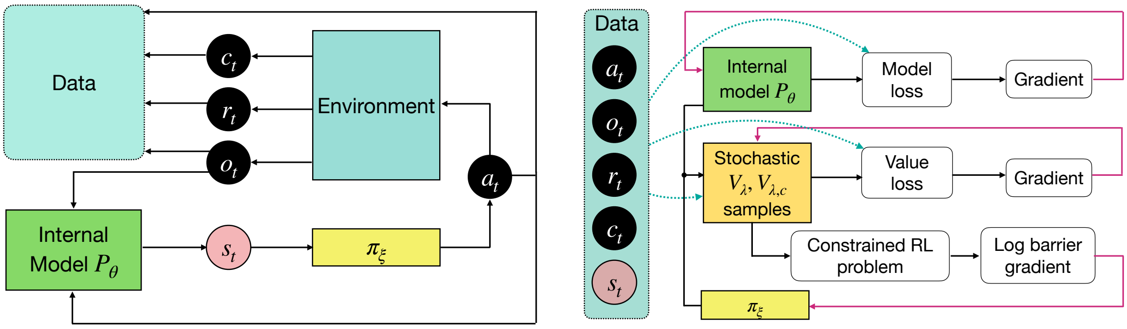


Figure 5.5: (a) Block diagram of collecting the data. (b) Block diagram of learning the models

Learning the policy We sample a sequence $\mathbf{s}_{\tau:\tau+H} \sim P_\theta(\mathbf{s}_{\tau:\tau+H})$, utilizing π_ξ to generate actions on the fly. By sampling B starting states s_τ and generating the sequences $\{s_t\}_{t=\tau}^{\tau+H}$ with the fixed P_θ, and using the fixed v^π_ψ, $v^\pi_{\psi_c}$, for policy π_ξ we approximate the stochastic inexact models of $J(\pi_\xi|P)$, $J_c(\pi_\xi|P)$ as follows

$$\tilde{J}(\pi_\xi|P_\theta, v^\pi_\psi) = \frac{1}{B}\sum_{\tau\in\{\tau_B\}}\frac{1}{H}\sum_{t=\tau}^{\tau+H}\mathbf{V}_\lambda(s_t), \tag{5.5}$$

$$\tilde{J}_c(\pi_\xi|P_\theta, v^\pi_{\psi_c}) = \frac{1}{B}\sum_{\tau\in\{\tau_B\}}\frac{1}{H}\sum_{t=\tau}^{\tau+H}\mathbf{V}_{\lambda,c}(s_t). \tag{5.6}$$

We get the gradient of the above loss over ξ using the back-propagation. We back-propagate through P_θ using path-wise gradient estimators [Moh+20].

In our past work LAMBDA, the constrained optimization for training the policy is done by using the Augmented Lagrangian approach. One can find the detailed description of this experiment and the approach in As, Usmanova, Curi, and Krause [As+22]. Here, would like to satisfy safety constraints not only by the end of the training, but also during the learning process. To increase safety during the learning we propose to use LB-SGD.

In particular, we replace the constrained optimization problem with an unconstrained approximation

$$\mathcal{L}_\pi(\pi_\xi) = \mathbb{E}\left[-\tilde{J}(\pi_\xi|P_\theta, v_\psi^\pi) - \eta \log(d - \tilde{J}_c(\pi_\xi|P_\theta, v_{\psi_c}^\pi))\right]. \tag{5.7}$$

Unfortunately, it is hard to verify that the noise in the objective and the value satisfy our assumptions. That is why we construct a heuristic upper (optimistic) confidence bound on J by sampling a batch of trajectories and taking maximum over them: $\hat{J}(\xi) = \max_{\theta\sim P_\Theta} \tilde{J}(\xi|P_\theta, v_\psi^\pi)$, and the heuristic upper (pessimistic) confidence bound on J_c: $\hat{J}_c(\xi) = \max_{\theta\sim P_\Theta} \tilde{J}_c(\xi|P_\theta, v_\psi^\pi)$ over the sampled trajectories. The log barrier gradient update of the policy parameter becomes

$$\xi \leftarrow \xi - \gamma(\xi)\nabla_\xi \mathcal{L}_\pi(\pi_\xi). \tag{5.8}$$

We use the step size $\gamma(\xi)$ as defined in the previous chapter, where the smoothness parameters are estimated empirically. Additionally, in case if the violation happens we use a heuristic allowing to go back to the feasible set. To do so, we remove the first part corresponding to the reward from the log barrier loss expression in Eq. (5.7) until we have the feasible policy again.

5.3.3 Experiments

Addressing the assumptions Let us briefly discuss the assumptions in Section 4.1 and explicitly state which of them do not hold. **Oracle.** As mentioned before, we cannot guarantee the assumptions in Section 2.1. LAMBDA uses neural networks to model the transition density and to learn an approximation of the objective and constraints[1]. For this reason, the assumptions on unbiased zeroth-order information and the variance of the oracles do not necessarily hold. **Smoothness.** However, by choosing ELU activation function [CUH15] we ensure the smoothness of our approximation of the objective and constraints. **MFCQ.** In general, similarly to the assumptions on the oracle, this cannot be guaranteed. However, in our experiments, the CMDP is defined to have only one constraint ($m = 1$) so this assumption is satisfied by definition. **Safe initial policy.** This assumption exists in a large body of previous work [Ber+17; Kol+18; WZ21]. Yet, it is not always clear how design such a policy a-priori.

Experiment protocol To ensure LB-SGD starts from a safe policy, we warm-start it with a policy that was trained on a similar, but easier task. Specifically, we follow a similar experimental setup as As, Usmanova, Curi, and Krause [As+22] but first train the

[1] The approximation of the objective and constraint is done by learning their corresponding *value functions*. Please see As, Usmanova, Curi, and Krause [As+22] for further details.

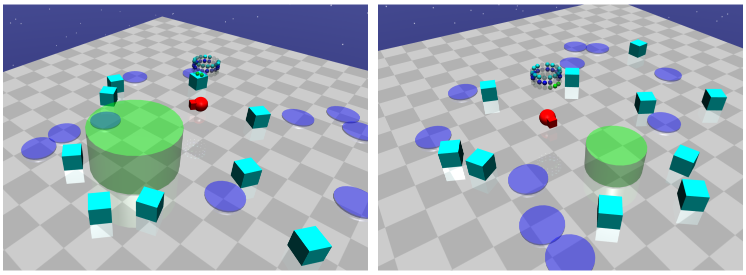


Figure 5.6: Starting from an easier task and continuing to a harder one. The robot on the righ picture should arrive to a smaller goal region, making navigation harder.

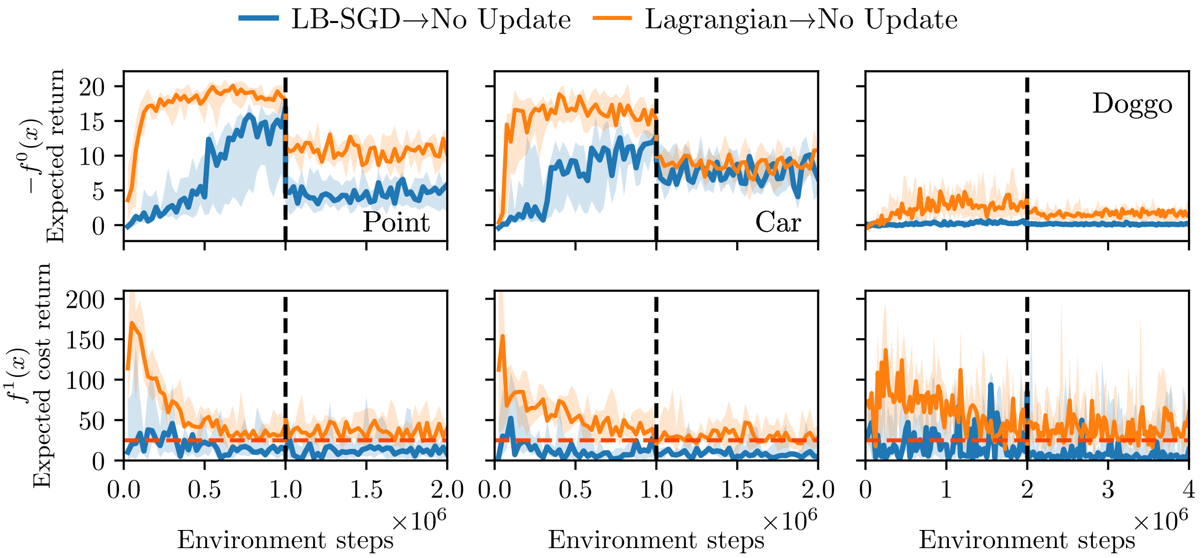


Figure 5.7: Across all different robots, LAMBDA with LB-SGD and the Augmented Lagrangian transfer well to the second stage in terms of safety. Since we do not update the policy, LAMBDA fails to reach the same task performance on the second stage.

agent with LAMBDA on a task in which the goal area is larger, as shown in Figure 5.6. We use the policy parameters of the trained agent as a starting point to LB-SGD on the harder task. As we later show, this allows the agent to start the second stage with a *safe but sub-optimal policy.* Beyond making sure that the assumption of an initially safe policy is fulfilled, we motivate this setup with the problem of *safe transfer learning.* In safe transfer learning, we want an agent to *safely adapt* to new tasks that share structure with previously-seen tasks. We verify this setup with all three available robots of the Safety-Gym benchmark suite [RAA19], each run with 5 different random seeds.

Results We first validate that LAMBDA's policy is safe but sub-optimal on the second stage of training. In Figure 5.7 we demonstrate how by using either LB-SGD or the Augmented Lagrangian on the first stage, and *not updating* the policy on the second stage, LAMBDA's policy is *safe* but *sub-optimal.* Further, given a safe and sub-optimal initial policy, we compare LB-SGD with the Augmented Lagrangian. Figure 5.8 demonstrates LB-SGD's ability to maintain the policy safe after transitioning to the new task. The

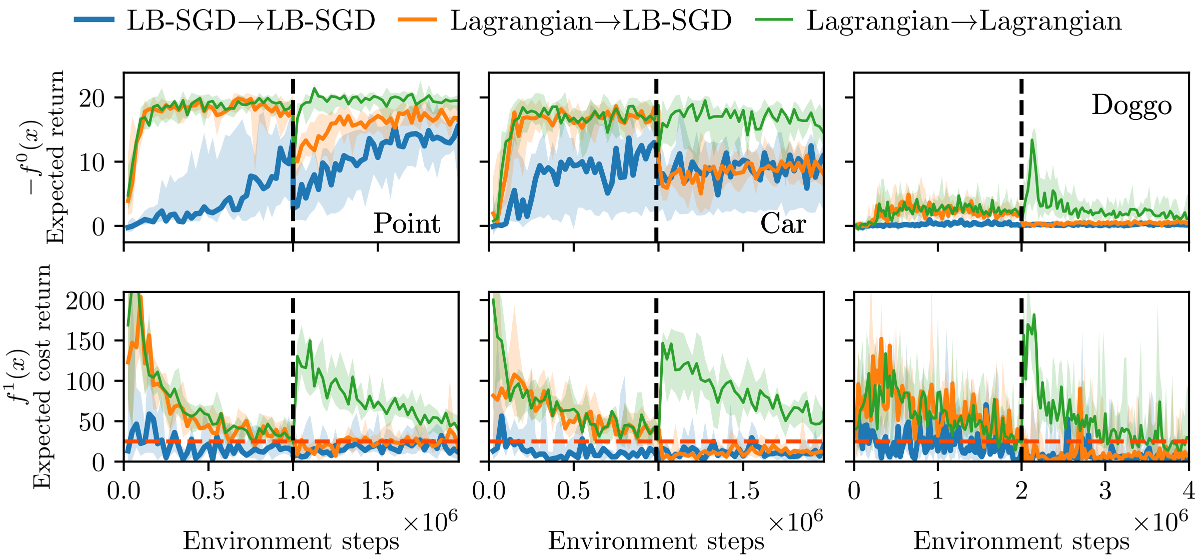


Figure 5.8: LAMBDA with LB-SGD transfers safely to the second task. The main trade-off, however, is the lower asymptotic performance of LB-SGD. Conversely, Lagrangian → Lagrangian fails transfer safely as the constraints with all robots rapidly grow when the new task is revealed (as shown by the vertical dashed black line).

Augmented Lagrangian needs to "re-learn" a new value for the Lagrange multiplier and therefore fails to transfer safely to the harder task. It is important to note that this safe transfer comes at the cost of limited exploration. As shown in Figures 5.7 and 5.8, LB-SGD reaches to less performant policies compared to the Augmented Lagrangian.

CHAPTER 6

Conclusion

In this thesis, we attempted to understand what complexity hides behind the safety requirement of the learning process. The safe learning problem is getting more and more important with the achievement of technologies allowing us to learn from real-world interactions; however, this problem is not been fully explored yet. We tried to understand what we could do about this problem from the optimization point of view. For the SFW method for unknown polytopic constraints, it appears that there is almost no loss in the convergence rate compared to the case with known constraints. For the LB-SGD method we proposed for non-linear optimization, it appears that we have to pay an additional order of $O(\varepsilon^{-3})$ for safety compared to non-safe methods with known constraints. We empirically compared the behavior of our approaches on synthetic problems with other existing methods and demonstrated that our approaches are capable of solving the problems in higher dimensions, including reinforcement learning (RL) problems. Empirically, we can also clearly observe that our safe methods learn much slower than unsafe ones.

Limitations and future directions Although our method for general constraints has a drawback of slow convergence compared to unsafe methods, our method updates are very computationally cheap and can be potentially used even for real-time computations. However, it would be very curious to explore for potential future work if these methods can be improved in terms of sample complexity, perhaps under some stronger assumptions. Additionally, in this thesis, we focused on finding the approximate solution only by the end of optimization. It would be interesting to explore what could be the performance of our method in terms of cumulative regret, assuming that we have to pay for every trial. And finally, we focused on the static problems, when the objective function and constraints stay the same during optimization, only assuming that we have noisy stochastic measurements. It would be exciting to extend our work for dynamic problems when the constraints and the objective can change over time, potentially restricting the speed of their change. This direction could make our work even more applicable to real-world problems. As one can see, there is plenty of potential future direction that we hope will be explored in the nearest future.

Outlook To summarize, we proposed two different optimization-based approaches for addressing the safe learning problems with linear and non-linear constraints. For the linear constraints setting, we proposed to estimate the model of the constraint polytope given all the measurements collected during learning, iteratively update this model, and use it within the Frank-Wolfe-based method with carefully chosen step-sizes. We provide the convergence rate for convex objectives and then prove the safety of all iterates with high probability. For non-linear constraints, we propose to approximate the problem with the logarithmic barrier subproblems and iteratively solve them using stochastic gradient descent (SGD) with a carefully chosen adaptive step size for safety. We analyze its convergence rate for non-convex, convex and strongly-convex problems, with first-order and zeroth-order noisy feedback, for smooth and non-smooth problems. We demonstrated the performance of our safe learning approach in simulations and applications such as control problems and reinforcement learning.

Appendices

APPENDIX A

Proofs of Chapter 3

A.1 Proof of Fact 2

Proof. Recall that the safety set $S_t(\bar{\delta})$ after iteration t is defined by the following inequalities:

$$S_t(\bar{\delta}) = \left\{ x \in \mathbb{R}^d : \forall i \in [m] \ \left[[\hat{a}_t^i]^T x - \hat{b}_t^i \right] + \phi^{-1}(\bar{\delta}/m)\sigma \left\| (\bar{X}_t^T \bar{X}_t)^{-1/2} \begin{bmatrix} x \\ -1 \end{bmatrix} \right\| \leq 0 \right\}. \tag{A.1}$$

Remember that $\bar{x}_t = \frac{X^T \mathbf{1}}{N}$ is an average of the measured points. Using the inversion formula for a block matrix, we obtain

$$(\bar{X}_t^T \bar{X}_t)^{-1} = \begin{bmatrix} X_t^T X_t & -X_t^T \mathbf{1} \\ -\mathbf{1}^T X_t & N_t \end{bmatrix}^{-1} = \begin{bmatrix} R_t & R_t \bar{x}_t \\ \bar{x}_t^T R_t & \frac{1}{N_t} + \bar{x}_t^T R_t \bar{x}_t \end{bmatrix},$$

where

$$R_t = \left[X_t^T X_t - N_t \bar{x}_t \bar{x}_t^T \right]^{-1} = \left[\sum_{j=1}^{N_t} (x_{(j)} - \bar{x}_t)(x_{(j)} - \bar{x}_t)^T \right]^{-1}. \tag{A.2}$$

Let us denote by $\phi_{\bar{\delta}} = \sigma \phi^{-1}(\bar{\delta}/m)$ and by $\epsilon_t^i = \hat{b}_t^i - (\hat{a}_t^i)^T x_t$. Then, the i-th inequality in Eq. (A.1)can be rewritten as follows:

$$\sqrt{\frac{\phi_{\bar{\delta}}^2}{N_t} + \phi_{\bar{\delta}}^2 (x - \bar{x}_t)^T R_t (x - \bar{x}_t)} \leq \epsilon_t^i.$$

Substituting $x = x_t$ to the above and combining the inequalities together, we obtain that the condition $x_t \in S_t(\bar{\delta})$ is equivalent to

$$\phi_{\bar{\delta}} \sqrt{\frac{1}{N_t} + (x_t - \bar{x}_t)^T R_t (x_t - \bar{x}_t)} \leq \min_{i \in [m]} \epsilon_t^i.$$

∎

A.2 Proof of Proposition 1

A.2.1 Important lemma for the proofs of Proposition 1 and Lemma 1

Lemma 14 defined in this section shows that each vertex of the approximated set $\hat{\mathcal{X}}_t$ differs from the corresponding vertex of the true set $\mathcal{X}$ by distance of order $O\left(\frac{1}{\sqrt{N_t}}\right)$. Let us recall that for the polytope $\mathcal{X} \in \mathbb{R}^d$, by an active set B we denote a set of indices of d linearly independent constraints active in a vertex $V^B \in \mathbb{R}^d$ of $\mathcal{X}$, i.e., $V^B = [A^B]^{-1}b^B$. Here, A^B is a corresponding sub-matrix of A and b^B is the corresponding right-hand-side of the constraint. The vertex estimate $\check{V}_t^B$ of a polytope is described by the system of linear equations $\hat{A}_t^B x = \hat{b}_t^B$.

Lemma 14. *If $\beta \in \mathcal{E}_t(\bar{\delta})$ and $N_t \geq \frac{C_{\bar{\delta}}^2}{(D_0+1)^2}$, where $C_{\bar{\delta}}$ is defined in Eq. (3.11), then for any vertex V^B defined by the active set B and its estimate $\check{V}_t^B$ we have that the estimation error is bounded by*

$$\|\check{V}_t^B - V^B\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}},$$

where $C_{\bar{\delta}} = \frac{2\phi_{\bar{\delta}} d(D_0+1)}{\rho_{\min}[A^B]}\sqrt{\frac{D_0^2+1}{\nu^2}+1}$.

Proof. Since the LSE (Least Squares Estimation) is unbiased,

$$\mathbb{E}\hat{A}_t^B = A^B, \; \mathbb{E}\hat{b}_t^B = b^B.$$

Let us denote by $\zeta_t = \hat{b}_t^B - b^B$ the uncertainty in estimation of b^B, and $G_t = \hat{A}_t^B - A^B$ the uncertainty in estimation of A^B.

Our aim is to bound the error of the vertex estimation $\|\check{V}_t^B - V^B\|$. Recall that

$$\check{V}_t^B - V^B = [\hat{A}_t^B]^{-1}\hat{b}_t^B - [A^B]^{-1}b^B = [A^B + G_t]^{-1}(b^B + \zeta_t) - [A^B]^{-1}b^B.$$

Note that for any matrices A, B it holds that

$$(A+B)^{-1} = A^{-1} - (I + A^{-1}B)^{-1}A^{-1}BA^{-1}.$$

Therefore, we can modify the expression for the $\check{V}_t^B - V^B$ as follows

$$\begin{aligned}
\check{V}_t^B - V^B &= \left[[A^B]^{-1} - (I + [A^B]^{-1}G_t)^{-1}[A^B]^{-1}G_t[A^B]^{-1}\right](b^B + \zeta_t) - [A^B]^{-1}b^B \\
&= [A^B]^{-1}b^B + [A^B]^{-1}\zeta_t - (I + [A^B]^{-1}G_t)^{-1}[A^B]^{-1}G_t[A^B]^{-1}(b^B + \zeta_t) - [A^B]^{-1}b^B \\
&= [A^B]^{-1}\zeta_t - (I + [A^B]^{-1}G_t)^{-1}[A^B]^{-1}G_t[A^B]^{-1}(b^B + \zeta_t).
\end{aligned}$$

The norm of the difference between the vertex V^B of the set $\mathcal{X}$ and its estimation can be bounded by

$$\|\hat{V}_t^B - V^B\| \leq \underbrace{\|[A^B]^{-1}\zeta_t\|}_{(a)} + \underbrace{\|[A^B]^{-1}G_t\|}_{(b)} \underbrace{\|V^B + [A^B]^{-1}\zeta_t\|}_{(c)} \underbrace{\| \left(I + [A^B]^{-1}G_t\right)^{-1} \|}_{(d)}. \quad \text{(A.3)}$$

To obtain the bounds on the terms (a),(b),(c),(d), let us first obtain the bounds on $\|G_t\|$ and $\|\zeta_t\|$.

Assume that for each $i \in [m]$ $\beta^i \in \mathcal{E}_t^i(\bar{\delta})$, where

$$\mathcal{E}_t^i(\bar{\delta}) = \left\{ z \in R^{d+1} : (\hat{\beta}_t^i - z)^T \Sigma_t^{-1} (\hat{\beta}_t^i - z) \leq \phi^{-1}(\bar{\delta})^2 \right\},$$

i.e., that for any active set B describing the vertex V^B we have $\beta^B \in \mathcal{E}_t^B(\bar{\delta})$. Consequently, $\|\hat{a}_t^i - a^i\|^2 + |\hat{b}^i - b^i|^2 \leq \phi^{-1}(\bar{\delta})\|\Sigma_t^{1/2}\|$ $\forall i \in B$. Then, for each row of G_t we have $\|\hat{a}_t^i - a^i\| \leq \phi^{-1}(\bar{\delta})\|\Sigma_t^{1/2}\|$, and for each element of ζ_t we have $|\hat{b}_t^i - b^i| \leq \phi^{-1}(\bar{\delta})\|\Sigma_t^{1/2}\|$. Hence, for $\|G_t\|$ we obtain

$$\|G_t\| \leq \|G\|_F = \sqrt{\sum_{i \in B} \|\hat{a}_t^i - a^i\|_2^2} \leq \sqrt{d}\phi^{-1}(\bar{\delta})\|\Sigma_t^{1/2}\|. \quad \text{(A.4)}$$

Similarly, we obtain a bound on $\|\zeta_t\|$:

$$\|\zeta_t\| = \sqrt{\sum_{i \in B} (\hat{b}_t^i - b^i)^2} \leq \sqrt{d}\phi^{-1}(\bar{\delta})\|\Sigma_t^{1/2}\|. \quad \text{(A.5)}$$

For the LSE covariance matrix norm $\|\Sigma_t^{1/2}\|$ we have

$$\begin{aligned} \|\Sigma_t^{1/2}\| = \sigma\|(\bar{X}_t^T \bar{X}_t)^{-1}\|^{1/2} &= \sigma \left\| \begin{bmatrix} I \\ \bar{x}_t^T \end{bmatrix} R_t \begin{bmatrix} I & \bar{x}_t \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1/N_t \end{bmatrix} \right\|^{1/2} \\ &\leq \sigma \sqrt{\|R_t\|\|\bar{x}_t\|^2 + \|R_t\| + \frac{1}{N_t}}, \end{aligned}$$

where R_t was defined in Eq. (A.2).

Note that we make measurements as it is described in Step 4 of the SFW algorithm, i.e., we make measurements at all coordinate directions within small step size ν from points generated by the method. Then, each new $2d$ measurements result in addition of a matrix $\sum_{j=N_t+1}^{N_t+2d}(x_{(j)} - \bar{x})(x_{(j)} - \bar{x})^T \succeq \nu^2 I$ to the matrix $R_t^{-1} = \sum_{j=1}^{N_t}(x_{(j)} - \bar{x}_t)(x_{(j)} - \bar{x}_t)^T$. Hence, $R_t^{-1} \succeq \frac{N_t \nu^2}{2d} I$. Hence, the minimal eigenvalue of the covariance matrix R_t^{-1} is bounded from below by the value $\lambda_{\min}(R_t^{-1}) \geq \frac{N_t \nu^2}{d}$. Thus, we obtain the following bound on the norm of R_t:

$$\|R_t\| \leq \frac{d}{N_t \nu^2}. \quad \text{(A.6)}$$

Note that $\bar{x}_t \in \mathcal{X}$, hence $\|\bar{x}_t\| \leq D_0$. It follows that

$$\begin{aligned}\|\Sigma_t^{1/2}\| &\leq \sigma\sqrt{\|R_t\|\|\bar{x}_t\|^2 + \|R_t\| + \frac{1}{N_t}} \leq \sigma\sqrt{\frac{d}{N_t\nu^2}\|\bar{x}_t\|^2 + \frac{d}{N_t\nu^2} + \frac{1}{N_t}} \\ &\leq \frac{\sigma\sqrt{d}\sqrt{\frac{D_0^2+1}{\nu^2} + \frac{1}{d}}}{\sqrt{N_t}} \leq \frac{\sigma\sqrt{d}\sqrt{\frac{D_0^2+1}{\nu^2} + 1}}{\sqrt{N_t}}. \qquad \text{(A.7)}\end{aligned}$$

In order to bound terms (a),(b),(c),(d) in Eq. (A.3), let us also bound the norm of the matrix $[A^B]^{-1}$:

$$\|[A^B]^{-1}\| = \rho_{\max}([A^B]^{-1}) = \frac{1}{\rho_{\min}[A^B]} \leq \frac{1}{\rho_{\min}(\mathcal{X})}. \qquad \text{(A.8)}$$

Then, combining inequalities (A.4),(A.5),(A.7),(A.8), we bound terms (a) and (b) as follows:

$$\begin{aligned}(a) : \left\|[A^B]^{-1}\zeta_t\right\| &\leq \|[A^B]^{-1}\|\|\zeta_t\| \leq \frac{U}{\sqrt{N_t}}, \\ (b) : \left\|[A^B]^{-1}G_t\right\| &\leq \|[A^B]^{-1}\|\|G_t\| \leq \frac{U}{\sqrt{N_t}},\end{aligned}$$

where U is defined by

$$U = \frac{\phi_{\bar{\delta}} d}{\rho_{\min}(\mathcal{X})}\sqrt{\frac{D_0^2+1}{\nu^2} + 1}.$$

Further, let us bound term (d). For $N_t \geq 4U^2$ it holds that

$$\begin{aligned}\left\|[A^B]^{-1}G_t\right\| &\leq \frac{U}{\sqrt{N_t}} \leq \frac{1}{2}, \\ \left\|[A^B]^{-1}\zeta_t\right\| &\leq \frac{U}{\sqrt{N_t}} \leq \frac{1}{2}.\end{aligned}$$

As such, for $N_t \geq 4U^2$ we have

$$\begin{aligned}\|\left(I + [A^B]^{-1}G_t\right)^{-1}\| &= \|I^{-1} - \left(I + [A^B]^{-1}G_t\right)^{-1}[A^B]^{-1}G_t\| \\ &\leq \|I\| + \|[A^B]^{-1}G_t\|\|\left(I + [A^B]^{-1}G_t\right)^{-1}\| \\ &\leq 1 + \frac{1}{2}\|\left(I + [A^B]^{-1}G_t\right)^{-1}\|.\end{aligned}$$

Hence

$$\left\|\left(I + [A^B]^{-1}G_t\right)^{-1}\right\| \leq 2. \qquad \text{(A.9)}$$

Finally, term (c) can be bounded as follows:

$$\left\|V^B + [A^B]^{-1}\zeta_t\right\| \leq D_0 + \frac{U}{\sqrt{N_t}}. \qquad \text{(A.10)}$$

Combining these all together, we obtain

$$\|\hat{V}_t^B - V^B\| \leq \frac{U}{\sqrt{N_t}} + 2\frac{U}{\sqrt{N_t}}\left(D_0 + \frac{U}{\sqrt{N_t}}\right) \leq 2\frac{U}{\sqrt{N_t}}(D_0+1) = \frac{C_{\bar{\delta}}}{\sqrt{N_t}},$$

where

$$C_{\bar{\delta}} = 2U(D_0+1) = \frac{2\phi_{\bar{\delta}} d(D_0+1)}{\rho_{\min}(\mathcal{X})}\sqrt{\frac{D_0^2+1}{\nu^2}+1}.$$

Since $N_t = C_n t^2(\ln t^2) \geq C_n$, the above bound holds under the proper choice of the constant

$$C_n \geq 4U^2 = \frac{C_{\bar{\delta}}^2}{(D_0+1)^2} = \frac{4d^2\phi_{\bar{\delta}}^2}{\rho_{\min}^2(\mathcal{X})}\left(\frac{D_0^2+1}{\nu^2}+1\right).$$

■

A.2.2 Proof of Proposition 1

Proof. This proof uses the result of Lemma 14 defined and proved in Section A.2.1. Let us bound the difference between the solution of the estimated DFS and the solution of the true DFS. Estimated DFS is a linear program defined by

$$\hat{s}_t = \arg\min_{\hat{A}_t x \leq \hat{b}_t} \langle c_t, x\rangle,$$

where $c_t = \nabla f(x_t)$. Any solution of such a linear program is a vertex (or convex hull of vertices) of the polytope $\hat{A}_t x \leq \hat{b}_t$. Let $\hat{V}_t^2$ be the estimated DFS solution vertex $\hat{V}_t^2 = \hat{s}_t = \arg\min_{s\in\hat{\mathcal{X}}_t}\langle c_t, s\rangle$. And correspondingly, let V_t^1 be the true DFS solution $V_t^1 = s_t = \arg\min_{s\in\mathcal{X}}\langle c_t, s\rangle$.

Let us define by $\Pi_{\hat{\mathcal{X}}_t} V_t^1$ the projection of V_t^1 onto $\hat{\mathcal{X}}_t$: $\bar{V}_t^1 = \Pi_{\hat{\mathcal{X}}_t} V_t^1$, and correspondingly we define $\tilde{V}_t^2$ as $\tilde{V}_t^2 = \Pi_{\mathcal{X}}\hat{V}_t^2$. Recall that the estimate $\check{V}^B$ of any vertex $V^B = [A^B]^{-1}b^B$ of the polytope $\mathcal{X}$ is described by the system of linear equations $\hat{A}_t^B x = \hat{b}_t^B$. Since the estimates $\check{V}^B$ denote intersections of the hyper-planes $\langle \hat{a}^j, s\rangle = \hat{b}^j\ \forall j \in B$ for some particular subset of indices B, the polytope $\hat{\mathcal{X}}_t$ lies inside the convex hull of the estimates $\check{V}_t^B$. Hence, any point $s \in \hat{\mathcal{X}}_t$ cannot be further from $\mathcal{X}$ than the estimates of all the vertices $\check{V}^i$ from $\mathcal{X}$. By Lemma 14, if $N_t \geq \frac{C_{\bar{\delta}}^2}{(D_0+1)^2}$ with $C_{\bar{\delta}}$ defined in Eq. (3.11) and $\beta \in \mathcal{E}_t(\bar{\delta})$, then for any vertex V^B of $\mathcal{X}$ we have $\|V^B - \check{V}_t^B\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}}$, thus we obtain that the distance from any point $s \in \hat{\mathcal{X}}_t$ to $\mathcal{X}$ is less than $\frac{C_{\bar{\delta}}}{\sqrt{N_t}}$. I.e., we have $\|\bar{V}_t^1 - V_t^1\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}}$.

Similarly, we show that the distance from any point $s \in \mathcal{X}$ to the set $\hat{\mathcal{X}}_t$ is upper bounded by $\frac{C_{\bar{\delta}}}{\sqrt{N_t}}$. Again, we can see that $\mathcal{X}$ is bounded by the convex hull of $V^{\hat{B}}$, where each $V^{\hat{B}} = [A^{\hat{B}}]^{-1}b^{\hat{B}}$ corresponds to a vertex $\check{V}_t^{\hat{B}} = [\hat{A}_t^{\hat{B}}]^{-1}\hat{b}_t^{\hat{B}}$ of $\hat{\mathcal{X}}_t$. Hence , the distance from any point $s \in \mathcal{X}$ to the set $\hat{\mathcal{X}}_t$ is upper bounded by $\max_{\hat{B}}\|V^{\hat{B}} - \check{V}_t^{\hat{B}}\|$. Then, by Lemma 14 we have $\|\tilde{V}_t^2 - \hat{V}_t^2\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}}$.

Note that $\bar{V}_t^1 \in \hat{\mathcal{X}}_t$, $\tilde{V}_t^2 \in \mathcal{X}$. From the definitions of $\hat{V}_t^2, V_t^1$ above it follows that

$$\begin{aligned} c_t^T \hat{V}_t^2 &\leq c_t^T \bar{V}_t^1, \\ c_t^T V_t^1 &\leq c_t^T \tilde{V}_t^2. \end{aligned}$$

Hence, we have

$$\begin{aligned} c_t^T \hat{V}_t^2 - \|c_t\| \frac{C_{\bar{\delta}}}{\sqrt{N_t}} &\leq c_t^T \tilde{V}_t^1 - \|c_t\| \frac{C_{\bar{\delta}}}{\sqrt{N_t}} \\ &\leq c_t^T V_t^1 \leq c_t^T \tilde{V}_t^2 \leq c^T \hat{V}_t^2 + \|c_t\| \frac{C_{\bar{\delta}}}{\sqrt{N_t}}. \end{aligned}$$

Thus, we obtain that if $\beta \in \mathcal{E}_t(\bar{\delta})$, then

$$E_t = |c_t^T(\hat{s}_t - s_t)| = |c_t^T \hat{V}_t^2 - c_t^T V_t^1| \leq \|c_t\| \frac{C_{\bar{\delta}}}{\sqrt{N_t}}.$$

Note that $\|c_t\| \leq M$, where M is the Lipschitz constant of the objective. Thus, if $\beta \in \mathcal{E}_t(\bar{\delta})$, then $E_t \leq \frac{C_{\bar{\delta}} M}{\sqrt{N_t}}$, i.e.,

$$\mathbb{P}\left\{ E_t \leq \frac{C_{\bar{\delta}} M}{\sqrt{N_t}} \right\} \geq 1 - \bar{\delta}.$$

■

A.3 Proof of Lemma 1

A.3.1 Supporting lemmas for the proof of Lemma 1

First, we provide some preliminary lemmas for the proof of Lemma 1. The proof of Lemma 1 also uses the result of Lemma 14 defined and proved in Section A.2.1. Let us denote by $\breve{x}_t = x_t - \bar{x}_t$, $\Delta_t^k = \hat{s}_t - x_k$, and recall that $\epsilon_t^i = \hat{b}_t^i - [\hat{a}_t^i]^T x_t$. By $\hat{V}_{t-1}$ we call the solution of the estimated DFS at the step $t-1$, by $\tilde{V}_{t-1} = \Pi_{\mathcal{X}} \hat{V}_{t-1}$, and by $\bar{V}_{t,t-1} = \Pi_{\hat{\mathcal{X}}_t} V_{t-1}$. By B_t we denote the active set corresponding to $\hat{V}_t$ (see Lemma 1 for definition) and by B_{t-1} the active set corresponding to $\hat{V}_{t-1}$

Lemma 15. *If $\beta \in \mathcal{E}_t(\bar{\delta}) \cap \mathcal{E}_{t-1}(\bar{\delta})$ holds, then we have*

$$\min_i \langle \hat{a}_t^i, \Delta_t^t \rangle \geq (1 - \gamma_{t-1}) \min_i \langle \hat{a}_t^i, \Delta_t^{t-1} \rangle - \frac{2\gamma_{t-1} \max_i \|\hat{a}_t^i\| C_{\bar{\delta}}}{\sqrt{N_{t-1}}}.$$

Proof.

$$\begin{aligned} \forall \hat{s}_t : \min_i \langle \hat{a}_t^i, \Delta_t^t \rangle &= \min_i \langle \hat{a}_t^i, \hat{s}_t - x_t \rangle = \min_{i \in B_t} \langle \hat{a}_t^i, \hat{s}_t - x_t \rangle = \min_{i \in B_t} \langle \hat{a}_t^i, \hat{s}_t - x_{t-1} - \gamma_{t-1}(\hat{s}_{t-1} - x_{t-1}) \rangle \\ &= \min_{i \in B_t} \langle \hat{a}_t^i, (1 - \gamma_{t-1})(\hat{s}_t - x_{t-1}) + \gamma_{t-1}(\hat{s}_{t-1} - \hat{s}_t) \rangle \end{aligned}$$

Denoting $j = \arg\min_{i \in B_t} \langle \hat{a}_t^i, \Delta_t^{t-1} \rangle)$, we get:

$$\forall \hat{s}_t : \min_i \langle \hat{a}_t^i, \Delta_t^t \rangle = (1 - \gamma_{t-1}) \langle \hat{a}_t^j, \hat{s}_t - x_{t-1} \rangle + \gamma_{t-1} \langle \hat{a}_t^j, \hat{s}_{t-1} - \hat{s}_t \rangle.$$

For the point s_t we have $\langle \hat{a}_t^j, s_t \rangle = 0$ and for any point $s \in \hat{\mathcal{X}}_t$ we have $\langle \hat{a}_t^j, s \rangle \geq 0$. Note that $\hat{V}_{t,t-1} \in \hat{\mathcal{X}}_t$ and that $\|\hat{V}_{t-1} - \hat{V}_{t,t-1}\| \leq \|\hat{V}_{t-1} - V_{t-1}\| + \|V_{t-1} - V_{t,t-1}\|$.

$$\begin{aligned} \min_i \langle \hat{a}_t^i, \Delta_t^t \rangle &\geq (1 - \gamma_{t-1}) \langle \hat{a}_t^j, \hat{s}_t - x_{t-1} \rangle + \gamma_{t-1} \langle \hat{a}_t^j, \hat{V}_{t-1} \rangle \\ &\geq (1 - \gamma_{t-1}) \langle \hat{a}_t^j, \hat{s}_t - x_{t-1} \rangle + \gamma_{t-1} \langle \hat{a}_t^j, \hat{V}_{t-1} - \hat{V}_{t,t-1} \rangle \\ &\geq (1 - \gamma_{t-1}) \min_i \langle \hat{a}_t^i, \hat{s}_t - x_{t-1} \rangle - \gamma_{t-1} \max_i \|\hat{a}_t^i\| \|\hat{V}_t - \hat{V}_{t,t-1}\|. \end{aligned} \tag{A.11}$$

Using the result of Lemma 14 we can obtain that if $\beta \in \mathcal{E}_t(\bar{\delta}) \cap \mathcal{E}_{t+1}(\bar{\delta})$ and $N_t \geq \frac{C_{\bar{\delta}}^2}{(D_0+1)^2}$ with $C_{\bar{\delta}}$ defined in Eq. (3.11), and applying the same arguments as in the proof of Proposition 1, we obtain

$$\|\hat{V}_{t-1} - \hat{V}_{t,t-1}\| \leq \|\hat{V}_{t-1} - V_{t-1}\| + \|V_{t-1} - \hat{V}_{t,t-1}\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}} + \frac{C_{\bar{\delta}}}{\sqrt{N_{t+1}}}. \tag{A.12}$$

Then, combining Eq. (A.12) with Eq. (A.11), we have

$$\min_i \langle \hat{a}_t^i, \Delta_t^t \rangle \geq (1 - \gamma_{t-1}) \min_i \langle \hat{a}_t^i, \Delta_t^{t-1} \rangle - \frac{2\gamma_{t-1} \max_i \|\hat{a}_t^i\| C_{\bar{\delta}}}{\sqrt{N_{t-1}}}.$$

■

Lemma 15 above is an induction step in the proof of Lemma 16. Lemma 16 below bounds the fastest rate of decreasing the distance to the boundaries of $\mathcal{X}$ for the SFW algorithm. Recall that $\mathcal{F}_t = \{\beta \in \cap_{k=0}^t \mathcal{E}_k(\bar{\delta})\}$.

Lemma 16. *If $\mathcal{F}_t$ holds, then we have*

$$\min_i \langle \hat{a}_t^i, \Delta_t^t \rangle \geq \frac{\min_i \langle \hat{a}_t^i, \Delta_t^0 \rangle}{t+2} \left(1 - \frac{C_{\bar{\delta}} \ln \ln t \max_i \|\hat{a}_t^i\|}{\sqrt{C_n} \min_i \langle \hat{a}_t^i, \bar{\Delta}_0 \rangle} \right).$$

Proof. By induction, from Lemma 15 we have

$$\min_i \langle \hat{a}_t^i, \hat{s}_t - x_t \rangle \geq \prod_{j=0}^{t-1} (1 - \gamma_j) \min_i \langle \hat{a}_t^i, x_0 - \hat{s}_t \rangle - \sum_{j=0}^{t-1} \frac{2C_{\bar{\delta}} \gamma_j}{\sqrt{N_j}} \max_i \|\hat{a}_t^i\| \prod_{k=j}^{t-1} (1 - \gamma_k).$$

Note that $1 - \gamma_k = \frac{k+1}{k+2}$, and

$$\prod_{k=j}^{t-1} (1 - \gamma_k) = \frac{(t)!/(j+1)!}{(t+1)!/(j+2)!} = \frac{j+2}{t+1}.$$

Thus,

$$\begin{aligned}\min_i \langle \hat{a}_t^i, \hat{s}_t - x_t\rangle &\geq \frac{1}{t+1}\min_i \langle \hat{a}_t^i, x_0 - \hat{s}_t\rangle - \sum_{j=0}^{t-1} \frac{j+2}{t+1}\frac{C_{\bar{\delta}}\gamma_j}{\sqrt{N_j}} \max_i \|\hat{a}_t^i\| \\ &= \frac{1}{t+1}\min_i \langle \hat{a}_t^i, x_0 - \hat{s}_t\rangle - \frac{1}{t+1}\sum_{j=0}^{t-1} \frac{C_{\bar{\delta}}}{\sqrt{N_j}} \max_i \|\hat{a}_t^i\|.\end{aligned}$$

Recall that $N_t = C_n t^2 (\ln t)^2$, hence we have

$$\begin{aligned}&\epsilon_t^i \geq \min_j \langle \hat{a}_t^j, \hat{s}_t - x_t\rangle \\ &\geq \frac{1}{t+2}\min_j \langle \hat{a}_t^j, \hat{s}_t - x_0\rangle - \frac{1}{t+2}\sum_{j=0}^{t} \frac{\sqrt{2\ln(j+1) + \ln 1/\bar{\delta}C_{\bar{\delta}}}}{\sqrt{C_n}(j+1)\ln(j+1)} \max_j \|\hat{a}_t^j\| = \\ &= \frac{1}{t+2}\left(\min_j \langle \hat{a}_t^j, \Delta_t^0\rangle - \frac{C_{\bar{\delta}}\ln(\ln t)}{\sqrt{C_n}} \max_j \|\hat{a}_t^j\|\right),\end{aligned}$$

where $\Delta_t^0 = \hat{s}_t - x_0$. ■

With Lemmas 15 and 16 in place, we are ready to prove Lemma 1.

A.3.2 Proof of Lemma 1

Proof. From Fact 2, the condition $x_t \in S_t(\bar{\delta})$ is equal to

$$\frac{\phi_{\bar{\delta}}^2}{N_t} + \phi_{\bar{\delta}}^2 (x_t - \bar{x}_t)^T R_t (x_t - \bar{x}_t) \leq \min_i [\epsilon_t^i]^2.$$

From the bound on $\|R_t\|$ given in Eq. (A.6) and knowing that D is a diameter of the set $\mathcal{X}$, we have

$$\frac{\phi_{\bar{\delta}}^2}{N_t} + \phi_{\bar{\delta}}^2 (x_t - \bar{x}_t)^T R_t (x_t - \bar{x}_t) \leq \frac{\phi_{\bar{\delta}}^2 \left(1 + \frac{dD^2}{\nu^2}\right)}{N_t}.$$

From Lemma 16 and recalling that $\epsilon_t^i = \min_i \langle \hat{a}_t^i, \Delta_t^t\rangle$, we have

$$[\epsilon_t^i]^2 \geq \frac{1}{(t+2)^2}\left(\min_i \langle \hat{a}_t^i, \Delta_t^0\rangle - \frac{C_{\bar{\delta}}\ln(\ln t)}{\sqrt{C_n}} \max_i \|\hat{a}_t^i\|\right)^2. \tag{A.13}$$

Hence, we can guarantee that $x_t \in S_t(\bar{\delta})$ if

$$N_t \geq \frac{(t+2)^2 \phi_{\bar{\delta}}^2 \left(1 + \frac{dD^2}{\nu^2}\right)}{\left(\min_i \langle \hat{a}_t^i, \Delta_t^0\rangle - \frac{C_{\bar{\delta}}\ln(\ln t)}{\sqrt{C_n}} \max_i \|\hat{a}_t^i\|\right)^2}. \tag{A.14}$$

We denote by $L_A = \max_i \|a_i\|$. Let us derive how far are $\min_i \langle \hat{a}_t^i, \Delta_t^0\rangle$ from ϵ_0 and

$\max_i \|\hat{a}_t^i\|$ from L_A. These are needed for obtaining a bound on the denominator above. Let us define by $\Delta_0 = s_t - x_0$, where s_t is the true vertex of $\hat{s}_t$ corresponding to $\min_i \langle \hat{a}_t^i, \Delta_t^0 \rangle$. Also recall that then $\min_i[\varepsilon_0] \leq \langle a^i, \Delta_0 \rangle$. If $C_n \geq \frac{C_{\bar{\delta}}^2}{(D_0+1)^2}$, then with probability greater than $1 - \bar{\delta}$ we have $\|\Delta_t^0 - \Delta_0\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}}$(using Lemma 14). We also can bound the difference $\|\hat{a}_t^i - a^i\|$ by $\|\hat{a}_t^i - a^i\| \leq \phi^{-1}(\bar{\delta}) \|\Sigma^{1/2}\| \leq \frac{C_{\bar{\delta}}}{\sqrt{N_t}} \frac{1}{\sqrt{d}\rho_{\min}(\mathcal{X})(D_0+1)}$. The second inequality follows from Eq. (A.7) and definition of $C_{\bar{\delta}}$ in Eq. (3.11).

Combining above inequalities together with the bound Eq. (A.14) on N_t we can conclude the following. If

$$C_n \geq \frac{4C_{\bar{\delta}}^2 (\ln \ln T)^2 L_A^2}{\min_i [\epsilon_0^i]^2},$$

then we can guarantee that $x_t \in S_t(\bar{\delta})$ by requiring

$$N_t \geq \frac{(t+2)^2 \phi_{\bar{\delta}}^2 \left(1 + \frac{dD^2}{\nu^2}\right)}{\min_i [\epsilon_0^i]^2}.$$

Since $n_t = C_n (t+1)(\ln(t+2))^2$ and $N_t = \sum_{k=0}^t n_k$, we obtain that

$$N_t \geq C_n (t+1)^2 (\ln(t+2))^2.$$

Hence, $C_n \geq \frac{\phi_{\bar{\delta}}^2 \left(1 + \frac{dD_0^2}{\nu^2}\right)}{(\ln(t+2))^2 \min_i [\epsilon_0^i]^2}$ is enough to ensure that $x_t \in S_t(\bar{\delta})$. Note that $C_{\bar{\delta}} \geq \phi_{\bar{\delta}}^2 \left(1 + \frac{dD^2}{\nu^2}\right)$. Thus, under the proper choice of constant parameter C_n, namely,

$$C_n \geq \max \left\{ \frac{4C_{\bar{\delta}}^2 (\ln \ln T)^2 L_A^2}{[\epsilon_0]^2}, \frac{C_{\bar{\delta}}^2}{(D_0 + 1)^2} \right\}$$

we conclude that $x_t \in S_t(\bar{\delta})$. ■

Remark Note that if we use a step size as in classical FW, $\gamma_t = \frac{2}{t+2}$, or in more general form $\gamma_t = \frac{l}{t+l}$ then we obtain that the distance to the boundaries $\min_i[\epsilon_t^i]$ will decrease with the rate upper bounded by $\prod_{k=0}^t (1 - \gamma_k) = \frac{l!}{t \cdot \ldots \cdot (t+l)} = O(\frac{1}{t^l})$ instead of Eq. (A.13) and this bound can be achieved e.g. in the case if the algorithm always moves in the same direction towards the boundary. This implies that in order to keep the convergence rate as in the original FW while satisfying $x_t \in S_t(\bar{\delta})$, due to Fact 2 we have to reduce the uncertainty of the boundaries faster, i.e., we need to take more measurements at each iteration.

A.4 Proof of Theorem 2

Proof. Let $\mu_t(\bar{\delta})$ denote a constant such that $\bar{E}_t(\bar{\delta}) = \frac{1}{2}\mu_t(\bar{\delta})\gamma_t C_f$. Then with probability $1-\bar{\delta}$ we have

$$\langle \hat{s}, \nabla f(x_t)\rangle \leq \min_{s\in\mathcal{X}}\langle s, \nabla f(x_t)\rangle + \frac{1}{2}\mu_t(\bar{\delta})\gamma_t C_f.$$

For the proof we refer to the following result from [Jag13]. This result holds in our setting since we use the same notions as in [Jag13] of g_t and s_t defined in Eq. (3.7).

Lemma 17. *(Lemma 5 [Jag13]) For a step $x_{t+1} = x_t + \gamma(\hat{s} - x_t)$ with an arbitrary step-size $\gamma \in [0,1]$, it holds that*

$$f(x_{t+1}) \leq f(x_t) - \gamma g_t + \frac{\gamma^2}{2}C_f(1+\mu_t),$$

if $\hat{s}$ is an approximate linear minimizer, i.e.

$$\langle \hat{s}, \nabla f(x_t)\rangle \leq \min_{\bar{s}\in\mathcal{X}}\langle \bar{s}, \nabla f(x_t)\rangle + \frac{1}{2}\mu_t\gamma C_f.$$

The step-size of the SFW algorithm is equal to $\gamma_t = \frac{1}{t+2}$. Let us define h_t as follows

$$h_t = h(x_t) = f(x_t) - f(x_*).$$

Then we obtain that

$$\begin{aligned} h_{t+1} &\leq h_t - \gamma_t g_t + \gamma_t^2\frac{C_f}{2}(1+\mu_t(\bar{\delta})) \leq h_t - \gamma_t h_t + \gamma_t^2\frac{C_f}{2}(1+\mu_t(\bar{\delta})) \\ &= (1-\gamma_t)h_t + \gamma_t^2\frac{C_f}{2}(1+\mu_t(\bar{\delta})). \end{aligned}$$

If we continue in the same manner, we obtain

$$\begin{aligned} h_{t+1} &\leq \prod_{i=0}^{t}(1-\gamma_i)h_0 + \sum_{k=0}^{t}\gamma_k^2\frac{C_f}{2}(1+\mu_k(\bar{\delta}))\prod_{i=k}^{t}(1-\gamma_i) \\ &= \prod_{i=0}^{t}\frac{i+1}{i+2}h_0 + \sum_{k=0}^{t}\frac{1}{(k+2)^2}\frac{C_f}{2}(1+\mu_k(\bar{\delta}))\prod_{i=k}^{t}\frac{i}{i+2} \\ &= \frac{1}{t+2}h_0 + \sum_{k=0}^{t}\frac{1}{(k+2)^2}\frac{C_f(1+\mu_k(\bar{\delta}))}{2}\frac{(t+1)!(k+2)!}{(t+2)!(k+1)!} \\ &= \frac{1}{t+2}\left(h_0 + \sum_{k=0}^{t}\frac{1}{(k+2)}\frac{C_f(1+\mu_k(\bar{\delta}))}{2}\right). \end{aligned}$$

Recall that $\bar{E}_t(\bar{\delta})$ denotes the upper bound on E_t with the confidence level $1-\bar{\delta}$. Due to

Proposition 1, we have

$$E_k(\bar{\delta}) = \frac{MC_{\bar{\delta}}}{\sqrt{N_k}} = \frac{MC_{\bar{\delta}}}{\sqrt{C_n}(k+2)\ln(k+2)}.$$

Hence, we obtain that

$$\mu_k(\bar{\delta}) = \frac{2E_k(\bar{\delta})(k+2)}{C_f} = \frac{2MC_{\bar{\delta}}}{C_f\sqrt{C_n}\ln(k+2)}.$$

Therefore, we finally obtain

$$h_{t+1} \leq \frac{h_0 + \ln(t+2)\frac{C_f}{2} + \sum_{k=0}^{t}\frac{C_f\mu_k(\bar{\delta})}{2}}{t+2} = \frac{h_0 + \ln(t+2)\frac{C_f}{2} + \ln\ln(t+2)\frac{C'}{2}}{t+2},$$

where $C' = \frac{MC_{\bar{\delta}}}{\sqrt{C_n}}$. ■

A.5 Proof of Corollary 1

Proof. Recall that

$$\phi^{-1}(\bar{\delta}) = \max\left\{\sqrt{128d\log N_t\log\left(\frac{N_t^2}{\bar{\delta}}\right)}, \frac{8}{3}\log\frac{N_t^2}{\bar{\delta}}\right\}.$$

Hence,

$$\phi_{\bar{\delta}} = O\left(\sigma\max\left\{\sqrt{d}\log t\sqrt{\log\frac{1}{\bar{\delta}}}, \log t + \log\frac{1}{\bar{\delta}}\right\}\right).$$

Recall that the total number of measurements N_t satisfies

$$N_t = 2C_{\bar{\delta}}^2\max\left\{\frac{4(\ln\ln T)^2L_A^2}{[\epsilon_0]^2}, \frac{1}{(D_0+1)^2}\right\}(t+2)^2(\ln(t+2))^2,$$

where $C_{\bar{\delta}} = \frac{2\phi_{\bar{\delta}}d(D_0+1)}{\rho_{\min}(\mathcal{X})}\sqrt{\frac{D_0^2+1}{\nu^2}+1}$. Thus, we conclude

$$N_t = \tilde{O}\left(\frac{\phi_{\bar{\delta}}^2 d^2}{t^2}\right) = \tilde{O}\left(\max\left\{\frac{d^3\ln\frac{1}{\bar{\delta}}}{\varepsilon^2}, \frac{d^2\ln^2\frac{1}{\bar{\delta}}}{\varepsilon^2}\right\}\right).$$

■

APPENDIX B

Proofs of Chapter 4

B.1 Proof of Lemma 2

Proof. Using the triangle inequality, we get

$$
\begin{aligned}
\|\Delta_t\| &= \|g_t - \nabla B_\eta(x_t)\| \\
&= \left\| G_n^0(x_t) - \nabla f^0(x_t) + \sum_{i=1}^{m} \left[\eta G_n^i(x_t) \left(\frac{1}{\bar{\alpha}_t^i} - \frac{1}{\alpha_t^i} \right) + \eta (G_n^i(x_t) - \nabla f^i(x_t)) \frac{1}{\alpha_t^i} \right] \right\| \\
&\leq \left\| G_n^0(x_t) - \nabla f^0(x_t) \right\| + \sum_{i=1}^{m} \left[\eta \left\| G_n^i(x_t) \right\| \left(\frac{1}{\bar{\alpha}_t^i} - \frac{1}{\alpha_t^i} \right) + \frac{\eta}{\alpha_t^i} \left\| G_n^i(x_t) - \nabla f^i(x_t)) \right\| \right].
\end{aligned}
$$

With high probability, we know $\|G_n^i(x_t)\| \leq L^i$, and from the sub-Gaussian property we have:

$$
\mathbb{P}\left\{ \left\| G_n^i(x_t) - \nabla f^i(x_t) \right\| \leq b_i + \hat{\sigma}_i(n) \sqrt{\ln \frac{1}{\delta}} \right\} \geq 1 - \delta
$$

$$
\mathbb{P}\left\{ \left| \alpha_t^i - \bar{\alpha}_t^i \right| \leq \sigma_i(n) \sqrt{\ln \frac{1}{\delta}} \right\} \geq 1 - \delta,
$$

from what we conclude:

$$
\mathbb{P}\left\{ \|\Delta_t\| \leq b_0 + \hat{\sigma}_0(n) \sqrt{\ln \frac{1}{\delta}} + \sum_{i=1}^{m} \frac{\eta}{\bar{\alpha}_t^i} \left(b_i + \hat{\sigma}_i(n) \sqrt{\ln \frac{1}{\delta}} \right) + \sum_{i=1}^{m} L_i \frac{\eta}{\alpha_t^i \bar{\alpha}_t^i} \sigma_i(n) \sqrt{\ln \frac{1}{\delta}} \right\} \geq 1 - \delta.
$$

$\blacksquare$

Bias

Proof. Using $\mathbb{E}[XY] \leq \sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}$, we get

$$
\begin{aligned}
\|\mathbb{E}\Delta_t\| &= \|\mathbb{E}[g_t - \nabla B_\eta(x_t)]\| \\
&\leq \left\| \mathbb{E}[G^0(x_t) - \nabla f^0(x_t)] + \sum_{i=1}^{m} \mathbb{E}\left[\eta G^i(x_t) \left(\frac{1}{\bar{\alpha}_t} - \frac{1}{\alpha_t} \right) + \eta (G^i(x_t) - \nabla f^i(x_t)) \frac{1}{\alpha_t} \right] \right\|
\end{aligned}
$$

$$
\begin{aligned}
&= \left\| \sum_{i=1}^{m} \mathbb{E}\left[\eta G^i(x_t)\left(\frac{1}{\bar{\alpha}_t} - \frac{1}{\alpha_t}\right)\right] \right\| + b_t^0 + \sum_{i=1}^{m} \frac{\eta}{\alpha_t^i} b_t^i \\
&\le \sum_{i=1}^{m} \mathbb{E}\left[\left\|\eta G^i(x_t)\left(\frac{1}{\bar{\alpha}_t^i} - \frac{1}{\alpha_t^i}\right)\right\|\right] + b_t^0 + \sum_{i=1}^{m} \frac{\eta}{\alpha_t^i} b_t^i \\
&\le \sum_{i=1}^{m} \sqrt{\mathbb{E}[\eta^2 \|G^i(x_t)\|^2]\mathbb{E}\left[\frac{1}{(\bar{\alpha}_t^i)^4}\|\alpha_t^i - \bar{\alpha}_t^i\|^2\right]} + b_t^0 + \sum_{i=1}^{m} \frac{\eta}{\alpha_t^i} b_t^i \\
&\le \sum_{i=1}^{m} \frac{\eta L_i \sigma_t}{(\alpha_t^i)^2} + \hat{b}_t^0 + \sum_{i=1}^{m} \frac{\eta}{\alpha_t^i} \hat{b}_t^i,
\end{aligned}
$$

■

B.2 Proof of the adaptivity

Proof. For adaptivity, we require

$$
f^i(x_{t+1}) \le \frac{f^i(x_t)}{2}.
$$

Using M^i-smoothness, we can bound the i-th constraint growth:

$$
\begin{aligned}
f^i(x_{t+1}) &\le f^i(x_t) + \langle \nabla f^i(x_t), x_{t+1} - x_t \rangle + \frac{M^i}{2}\|x_t - x_{t+1}\|_2^2 \\
&= f^i(x_t) - \gamma_t \langle \nabla f^i(x_t), g_t \rangle + \gamma_t^2 \frac{M^i}{2}\|g_t\|_2^2
\end{aligned}
$$

That is, the condition on γ_t for adaptivity (and safety) we can formulate by

$$
-\gamma_t \langle \nabla f^i(x_t), g_t \rangle + \gamma_t^2 \frac{M^i}{2}\|g_t\|_2^2 \le \frac{-f^i(x_t)}{2} = \frac{\alpha_t^i}{2}.
$$

By carefully rewriting the above inequality without strengthening it, we get

$$
\begin{aligned}
\gamma_t^2 \frac{M^i}{2}\|g_t\|_2^2 - \gamma_t \langle \nabla f^i(x_t), g_t \rangle \pm \frac{1}{2M^i}\frac{\langle \nabla f^i(x_t), g_t \rangle^2}{\|g_t\|_2^2} &\le \frac{\alpha_t}{2} \\
\left(\gamma_t \|g_t\|_2 - \frac{\langle \nabla f^i(x_t), g_t \rangle}{M_i \|g_t\|_2}\right)^2 &\le \frac{\alpha_t}{M^i} + \frac{\langle \nabla f^i(x_t), g_t \rangle^2}{M_i^2 \|g_t\|_2^2}
\end{aligned}
$$

Using the quadratic inequality solution, we obtain the following sufficient bound on the adaptive γ_t :

$$
\gamma_t \|g_t\|_2 \le \frac{\langle \nabla f^i(x_t), g_t \rangle}{M_i \|g_t\|_2} + \sqrt{\frac{\alpha_t}{M^i} + \frac{\langle \nabla f^i(x_t), g_t \rangle^2}{M_i^2 \|g_t\|_2^2}} = (*)
$$

Then, we can rewrite this expression of the right part as follows:

$$(*) = \sqrt{\frac{\langle \nabla f^i(x_t), g_t\rangle^2}{M_i^2\|g_t\|_2^2} + \frac{\alpha_t^i}{M^i}} + \frac{\langle \nabla f^i(x_t), g_t\rangle}{M_i\|g_t\|_2} = \sqrt{\frac{\alpha_t^i}{M^i}}\left(\sqrt{\frac{\langle \nabla f^i(x_t), g_t\rangle^2}{M_i\alpha_t^i\|g_t\|_2^2} + 1} + \frac{\langle \nabla f^i(x_t), g_t\rangle}{\sqrt{M_i\alpha_t^i}\|g_t\|_2}\right)$$

$$= \sqrt{\frac{\alpha_t^i}{M^i}}\frac{1}{\sqrt{\frac{\langle \nabla f^i(x_t), g_t\rangle^2}{M_i\alpha_t^i\|g_t\|_2^2} + 1} - \frac{\langle \nabla f^i(x_t), g_t\rangle}{\sqrt{M_i\alpha_t^i}\|g_t\|_2}} = \frac{\alpha_t^i}{\sqrt{\frac{\langle \nabla f^i(x_t), g_t\rangle^2}{\|g_t\|_2^2} + M^i\alpha_t^i} - \frac{\langle \nabla f^i(x_t), g_t\rangle}{\|g_t\|_2}} = \frac{\alpha_t^i}{\sqrt{(\theta_t^i)^2 + M^i\alpha_t^i} - \theta_t^i}$$

Therefore, the condition $\gamma_t \leq \min_{i\in[1,m]}\left\{\frac{\alpha_t^i}{\sqrt{(\theta_t^i)^2+M^i\alpha_t^i}-\theta_t^i}\right\}\frac{1}{\|g_t\|_2}$ is sufficient for $f^i(x_{t+1}) \leq \frac{f^i(x_t)}{2}$. Using the Cauchy-Schwartz inequality, we can simplify this condition (but making it more conservative):

$$(*) \geq \frac{\alpha_t^i}{\sqrt{(\theta_t^i)^2 + \alpha_t M_i} + |\theta_t^i|} \geq \frac{\alpha_t^i}{2|\theta_t^i| + \sqrt{\alpha_t^i M^i}}.$$

■

B.3 Proof of the local smoothness

Proof. Let us define the hessian of the log-barrier $B_\eta(x)$ by $H_B(y)$ at the region $y \in U$ around x_t such that $x_{t+1} \in U$. Note that by definition of the log barrier, the hessian of it at the point x_{t+1} is given by

$$\nabla^2 B_\eta(x_t) = \nabla^2 f^0(x_{t+1}) + \sum_{i=1}^m \eta\frac{\nabla^2 f^i(x_{t+1})}{-f^i(x_{t+1})} + \sum_{i=1}^m \eta\frac{\nabla f^i(x_{t+1})\nabla f^i(x_{t+1})^T}{(-f^i(x_{t+1}))^2}.$$

Based on that,

$$\begin{aligned}|g_t^T H_B(x_{t+1})g_t| &\leq M^0\|g_t\|_2^2 + \eta\sum_{i=1}^m \frac{M^i}{\alpha_{t+1}^i}\|g_t\|_2^2 + \eta\sum_{i=1}^m \frac{(\nabla f^i(x_{t+1})^T g_t/\|g_t\|_2)^2}{(\alpha_{t+1}^i)^2}\|g_t\|_2^2\\ &\leq \|g_t\|_2^2\left(M^0 + \eta\sum_{i=1}^m \frac{M^i}{\alpha_{t+1}^i} + \eta\sum_{i=1}^m \frac{\langle \nabla f^i(x_{t+1}), g_t\rangle^2/\|g_t\|_2^2}{(\alpha_{t+1}^i)^2}\right)\\ &\leq \|g_t\|_2^2\left(M^0 + 2\eta\sum_{i=1}^m \frac{M^i}{\alpha_t^i} + 4\eta\sum_{i=1}^m \frac{\langle \nabla f^i(x_{t+1}), g_t\rangle^2}{(\alpha_t^i)^2\|g_t\|_2^2}\right)\end{aligned}$$

Thus,

$$M_2(x_t) = M^0 + 2\eta\sum_{i=1}^m \frac{M^i}{\alpha_t^i} + 4\eta\sum_{i=1}^m \frac{\langle \nabla f^i(x_{t+1}), g_t\rangle^2}{\alpha_t^2\|g_t\|_2^2}$$

$$M_2(x_t) \leq M^0 + 2\eta\sum_{i=1}^m \frac{M^i}{\alpha_t} + 4\eta\sum_{i=1}^m \frac{4(\theta_t^i)^2 + (\theta_t^i)^2 + M^i\alpha_t^i + 2\theta_t^i\sqrt{M^i\alpha_t^i}}{(\alpha_t^i)^2}$$

■

B.4 Proof of Fact 3

Proof. Using the local smoothness of the log barrier, we can see:

$$\begin{aligned}
\eta \sum_{i\in\mathcal{I}_t} -\log\alpha^i_{t+1} &\le \eta\sum_{i\in\mathcal{I}_t} -\log\alpha^i_t - \gamma_t\langle \eta\sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}, g_t\rangle + \frac{M_2(x_t)}{2}\gamma_t^2\|g_t\|^2 \\
&\le \eta\sum_{i\in\mathcal{I}_t} -\log\alpha^i_t + \gamma_t\left(-\langle \eta\sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}, g_t\rangle + \frac{1}{2}\|g_t\|^2\right) \\
&= \eta\sum_{i\in\mathcal{I}_t} -\log\alpha^i_t + \frac{\gamma_t\eta^2}{2}\left(2\langle A, A+B\rangle + \|A+B\|^2\right) \\
&= \eta\sum_{i\in\mathcal{I}_t} -\log\alpha^i_t + \frac{\gamma_t\eta^2}{2}\left(\|B\|^2 - \|A\|^2\right), \qquad \text{(B.1)}
\end{aligned}$$

where $g_t = A+B$, with $A := \sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}$ and $B := \frac{g_t}{\eta} - \sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}$. Using Assumption 4 we obtain a lower bound on $\|A\|$:

$$\|A\| = \left\|\sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}\right\| \ge \langle\sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}, s_x\rangle \ge \sum_{i\in\mathcal{I}_t}\frac{\langle\nabla f^i(x_t), s_x\rangle}{\alpha^i_t} \ge \sum_{i\in\mathcal{I}_t}\frac{l}{\alpha^i_t}. \qquad \text{(B.2)}$$

The second part $\|B\|$ we can upper bound with high probability $1-\delta$ as follows:

$$\begin{aligned}
\|B\| &= \left\|\frac{G^0_n(x_t,\xi_t)}{\eta} + \sum_{j\notin\mathcal{I}_t}\frac{G^j_n(x_t,\xi_t)}{\bar\alpha^j_t} + \sum_{i\in\mathcal{I}_t}\frac{G^i_n(x_t,\xi_t)}{\bar\alpha^i_t} - \sum_{i\in\mathcal{I}_t}\frac{\nabla f^i(x_t)}{\alpha^i_t}\right\| \qquad \text{(B.3)} \\
&\le \frac{L}{\eta}\left(1 + 2(m - |\mathcal{I}_t|)\right) + 2\sum_{i\in\mathcal{I}_t}\frac{\hat\sigma_i(n)\sqrt{\ln\frac{1}{\delta}}}{(\alpha^i_t)^2} \le \frac{L}{\eta}\left(2m+1\right), \qquad \text{(B.4)}
\end{aligned}$$

for $\hat\sigma_i(n) \le \frac{(\alpha^i_t)^2 L}{\eta\sqrt{\ln\frac{1}{\delta}}}$, and $\sigma_i(n) \le \frac{\alpha^i_t}{2\sqrt{\ln\frac{1}{\delta}}}$, implying $\bar\alpha^i_t \ge \alpha^i_t/2$. Then, if $\min\alpha^i_t \le \bar c\eta$, we have $\sum_{i\in\mathcal{I}_t}\frac{1}{\alpha^i_t} \ge \frac{1}{\bar c\eta} = \frac{L}{l\eta}\left(2m+1\right)$, and therefore with high probability $\|B\| \le \|A\|$. Then we get Eq. (B.1), that implies

$$\prod_{i\in\mathcal{I}_t}\alpha^i_{t+1} \ge \prod_{i\in\mathcal{I}_t}\alpha^i_t. \qquad \text{(B.5)}$$

Moreover, using the same reasoning, we can prove that

$$\prod_{i\in\mathcal{I}}\alpha^i_{t+1} \ge \prod_{i\in\mathcal{I}}\alpha^i_t. \qquad \text{(B.6)}$$

for any subset of indices $\mathcal{I} \subseteq [m]$ such that $\mathcal{I}_t \subseteq \mathcal{I}$. $\blacksquare$

B.5 Lower bound on γ_t

Here we assume $\underline{\alpha}_t^i \geq c\eta$. Recall that

$$\gamma_t = \min\left\{ \min_{i\in[m]} \left\{ \frac{\underline{\alpha}_t^i}{2|\hat{\theta}_t^i| + \sqrt{\underline{\alpha}_t^i M^i}} \right\} \frac{1}{\|g_t\|_2}, \frac{1}{\hat{M}_2(x_t)} \right\}.$$

where

$$\hat{M}_2(x_t) = M^0 + 6\eta \sum_{i=1}^m \frac{M^i}{\underline{\alpha}_t^i} + 20\eta \sum_{i=1}^m \frac{(\hat{\theta}_t^i)^2}{(\underline{\alpha}_t^i)^2}.$$

We get the lower bound by constructing a bound on both of the terms inside the minimum. 1) We have $\mathbb{P}\left\{ \hat{M}_2(x_t) \leq \left(1 + 6\frac{m}{c}\right) M + 20\frac{mL^2}{\eta c^2} \right\} \geq 1 - \delta$ (Due to Lemma 6, and by definition of $\hat{M}_2(x_t)$), which implies

$$\mathbb{P}\left\{ \frac{1}{\hat{M}_2(x_t)} \geq \eta \left(\frac{1}{\frac{20m}{c^2}L^2 + \eta(1 + 6\frac{m}{c})M} \right) \right\} \geq 1 - \delta.$$

2) Using Lemma 6 we get $\mathbb{P}\left\{ \|g_t\| \leq L^0 + \sum_{i=1}^m \frac{L^i}{c} \right\} \geq 1 - \delta$. Hence, we can bound

$$\mathbb{P}\left\{ \min_{i\in[m]} \left\{ \frac{\underline{\alpha}_t^i}{2|\hat{\theta}_t^i| + \sqrt{\underline{\alpha}_t^i M^i}} \right\} \frac{1}{\|g_t\|_2} \geq \frac{c\eta}{(2L + \sqrt{Mc\eta})L(1 + \frac{m}{c})} \right\} \geq 1 - \delta.$$

Therefore,

$$\mathbb{P}\left\{ \gamma_t \geq \frac{\eta}{2} \min\left\{ \frac{1}{\frac{10m}{c^2}L^2 + \eta(1 + 3\frac{m}{c})M}, \frac{1}{L^2(\frac{1}{c} + \frac{m}{c^2}) + 0.5\sqrt{\frac{M\eta}{cL^2}}L^2(1 + \frac{m}{c})} \right\} \right\} \geq 1 - \delta,$$

$$\mathbb{P}\left\{ \gamma_t \geq \frac{\eta}{2L^2(1 + \frac{m}{c})} \min\left\{ \frac{1}{\frac{10}{c} + \frac{3M\eta}{L^2}}, \frac{1}{\frac{1}{c} + \sqrt{\frac{M\eta}{4cL^2}}} \right\} \right\} \geq 1 - \delta.$$

$$\mathbb{P}\left\{ \gamma_t \geq \frac{c\eta}{2L^2(1 + \frac{m}{c})} \min\left\{ \frac{1}{10 + \frac{3Mc\eta}{L^2}}, \frac{1}{1 + \sqrt{\frac{Mc\eta}{4L^2}}} \right\} \right\} \geq 1 - \delta.$$

Finally, the bound is

$$\mathbb{P}\left\{ \gamma_t \geq \eta C \right\} \geq 1 - \delta.$$

with

$$C := \frac{c\eta}{2L^2(1+\frac{m}{c})} \min\left\{\frac{1}{10+\frac{3Mc\eta}{L^2}}, \frac{1}{1+\sqrt{\frac{Mc\eta}{4L^2}}}\right\}.$$

B.6 Proof of Lemma 8

Proof. From Fact 4 it follows that

$$\forall x \in \mathcal{X}\ \exists s_x = \frac{x-x_0}{\|x-x_0\|} \in \mathbb{R}^d :\ \langle s_x, \nabla f^i(x)\rangle \geq \frac{\beta}{2D} \quad \forall i \in \mathcal{I}_{\beta/2}(x).$$

Let $\hat{x}$ be an approximately optimal point for the log barrier: $B_\eta(\hat{x}) - B_\eta(x_\eta^*) \leq \eta$, that is equivalent to:

$$f^0(\hat{x}) + \eta\sum_{i=1}^m -\log(-f^i(\hat{x})) - f^0(x_\eta^*) - \eta\sum_{i=1}^m -\log(-f^i(x_\eta^*)) \leq \eta.$$

Then, for the objective function we have the following bound:

$$f^0(\hat{x}) - f^0(x_\eta^*) \leq \eta + \eta\sum_{i=1}^m -\log\frac{-f^i(x_\eta^*)}{-f^i(\hat{x})}. \tag{B.7}$$

The optimal point for the log barrier x_η^* must satisfy the stationarity condition

$$\nabla B_\eta(x_\eta^*) = \nabla f^0(x_\eta^*) + \eta\sum_{i=1}^m \frac{\nabla f^i(x_\eta^*)}{-f^i(x_\eta^*)} = 0.$$

By carefully rearranging the above, we obtain

$$\sum_{i\in\mathcal{I}_{\beta/2}(x_\eta^*)} \frac{\nabla f^i(x_\eta^*)}{-f^i(x_\eta^*)} + \sum_{i\notin\mathcal{I}_{\beta/2}(x_\eta^*)} \frac{\nabla f^i(x_\eta^*)}{-f^i(x_\eta^*)} = \frac{-\nabla f^0(x_\eta^*)}{\eta}.$$

By taking a dot product of both sides of the above equation with $s_x = \frac{x_\eta^*-x_0}{\|x_\eta^*-x_0\|}$, using the Lipschitz continuity we get for x_η^*:

$$\frac{1}{\min_i\{-f^i(x_\eta^*)\}} \sum_{i\in\mathcal{I}_{\beta/2}(x_\eta^*)} \langle\nabla f^i(x_\eta^*), s_x\rangle \frac{\min_i\{-f^i(x_\eta^*)\}}{-f^i(x_\eta^*)} \tag{B.8}$$

$$= \frac{\langle -\nabla f^0(x_\eta^*), s_x\rangle}{\eta} - \sum_{i\notin\mathcal{I}_{\beta/2}(x_\eta^*)} \frac{\langle\nabla f^i(x_\eta^*), s_x\rangle}{-f^i(x_\eta^*)} \leq \frac{mL}{\eta}. \tag{B.9}$$

From the above, using Fact 4, we get

$$\min\{-f^i(x^*_\eta)\} \geq \frac{\eta\beta}{2mLD}.$$

Hence, combining the above with Eq. (B.7) we get the following relation of point $\hat{x}$ and point x^*_η optimal for the log barrier:

$$f^0(\hat{x}) - f^0(x^*_\eta) \leq \eta + \eta \sum_{i=1}^m \log \frac{-f^i(\hat{x})}{-f^i(x^*_\eta)} \leq \eta \left(1 + m \log \left(\frac{2mLD\hat{\beta}}{\eta\beta}\right)\right). \tag{B.10}$$

Next, note that the Lagrangian $\mathcal{L}(x, \lambda)$ is a convex function over x and concave over λ. Hence, for $(x^*_\eta, \lambda^*_\eta) := \left(x^*_\eta, \left[\frac{\eta}{-f^1(x^*_\eta)}, \dots, \frac{\eta}{-f^m(x^*_\eta)}\right]^T\right)$ we have

$$\mathcal{L}(x^*_\eta, \lambda^*_\eta) - \mathcal{L}(x^*, \lambda^*) \leq \mathcal{L}(x^*_\eta, \lambda^*_\eta) - \mathcal{L}(x^*, \lambda^*_\eta) \leq \langle \nabla_x \mathcal{L}(x^*_\eta, \lambda^*_\eta), \lambda^*_\eta - x^* \rangle \leq 0.$$

Expressing $\mathcal{L}(x^*_\eta, \lambda^*_\eta)$ and $\mathcal{L}(x^*, \lambda^*)$ and exploiting the fact that $\nabla B_\eta(x^*_\eta) = \nabla_x \mathcal{L}(x^*_\eta, \lambda^*_\eta) = 0$, we obtain $\mathcal{L}(x^*_\eta, \lambda^*_\eta) - \mathcal{L}(x^*, \lambda^*) = f^0(x^*_\eta) - f^0(x^*) - m\eta \leq 0$. Consequently, we have $f^0(x^*_\eta) - f^0(x^*) \leq m\eta$. Combining the above and Eq. (B.10), we get

$$f^0(\hat{x}) - \min_{x \in \mathcal{X}} f^0(x) \leq \eta + \eta m \log \left(\frac{2mLD\hat{\beta}}{\eta\beta}\right) + m\eta.$$

■

B.7 Zeroth-order estimator properties proof

The deviation of the gradient estimators $G^i(x_t, \nu) - \nabla f^i_\nu(x_t)$, by definition can be expressed as follows for $i = 0, \dots, m$

$$G^i(x_t, \nu) - \nabla f^i_\nu(x_t) = \frac{1}{n_t} \sum_{j=1}^{n_t} \left[\underbrace{\left(d \frac{f^i(x_k + \nu s_{tj}) - f^i(x_t)}{\nu} s_{tj} - \nabla f^i_\nu(x_t) \right)}_{v^i_j} + \underbrace{d \frac{\xi^{i+}_{tj} - \xi^{i-}_{tj}}{\nu} s_{tj}}_{u^i_j} \right], \tag{B.11}$$

where the first term under the summation v^i_j is dependent only on random s_{tj}, however the second term is dependent on both random variables coming from the noise $\xi^{i\pm}_{tj}$ and from the direction s_{tj}.

We use the result of Lemma 3.11 [Ber+21], we can bound v_j^i

$$\mathbb{E}\left\|\frac{1}{n}\sum_{j=1}^{n} v_j^i\right\|^2 \leq 3\frac{\frac{d}{d+2}\|\nabla f^i(x)\|^2 + dM_i^2\nu^2}{n}. \ \forall i \in \{0,\dots,m\}. \tag{B.12}$$

The second part u_j^i is zero-mean, hence does not influence the bias. Indeed, using the independence of $\xi_{tj}^{j\pm}$ and s_{tj} we derive

$$\mathbb{E}\sum_{j=1}^{n_t} u_j^i = \frac{d}{\nu}\mathbb{E}\left(\sum_{j=1}^{n_t}(\xi_{tj}^{i+} - \xi_{tj}^{i-})s_{tj}\right) = 0. \tag{B.13}$$

Its variance can be bounded as follows, using $\|s_{tj}\| = 1$:

$$\mathbb{E}\left\|\frac{1}{n}\sum_{j=1}^{n} u_j^i\right\|^2 = \mathbb{E}\frac{d^2}{\nu^2}\left\|\sum_{j=1}^{n}(\xi_{tj}^{i+} - \xi_{tj}^{i-})s_{tj}\right\|^2 \leq \frac{d^2\sigma^2}{\nu^2 n}. \tag{B.14}$$

From the above, and Lemma 3.11 the statement of the Lemma follows directly.

B.8 Proof of property 2 in Fact 5:

Proof. By definition and the property of the smoothed function $\nabla f_\nu(x) = \mathbb{E}_s d\frac{f(x+\nu s)-f(x)}{\nu}s$, where $s \sim \mathcal{U}(\mathcal{S}^d)$. Hence

$$\nabla f_\nu(x) - \nabla f_\nu(y) = \mathbb{E}_s\left[d\frac{f(x+\nu s) - f(x)}{\nu}s - d\frac{f(y+\nu s) - f(y)}{\nu}s\right] = d\mathbb{E}_s\frac{f(x+\nu s) - f(y+\nu s)}{\nu}s.$$

Let us denote by $\delta_f(s)$ the function:

$$\delta_f(s) := f(x+\nu s) - f(y+\nu s).$$

Then, we have:

$$\|\nabla f_\nu(x) - \nabla f_\nu(y)\|_2 = \frac{d}{\nu}\|\mathbb{E}_s\delta_f(s)s\|_2. \tag{B.15}$$

First, note that the absolute value of $\delta f(s)$ is bounded by

$$|\delta_f(s)| = |f(x+\nu s) - f(y+\nu s)| \leq L\|x-y\|_2.$$

Assume that $r \in \mathbb{R}^d : \|r\|_2 = 1$ is the unit vector of the direction of $\mathbb{E}_s\delta f(s)s$. Then,

$$\begin{aligned}\|\mathbb{E}_s\delta f(s)s\|_2 &= \langle\mathbb{E}_s\delta_f(s)s, r\rangle = \mathbb{E}_s\delta_f(s)\langle s, r\rangle \\ &= \frac{1}{2}\mathbb{E}_s[\delta_f(s)\langle s, r\rangle | \langle s, r\rangle \geq 0] + \frac{1}{2}\mathbb{E}_s[\delta_f(s)\langle s, r\rangle | \langle s, r\rangle < 0]\end{aligned}$$

$$= \frac{1}{2}\mathbb{E}_{s\in\mathcal{S}^d,\langle s,r\rangle\geq 0}[\delta_f(s)\langle s,r\rangle] + \frac{1}{2}\mathbb{E}_{s\in\mathcal{S}^d,\langle s,r\rangle\geq 0}[\delta_f(-s)\langle s,r\rangle].$$

Note that in the above terms the multiplicands $\langle s,r\rangle$ are positive. Therefore, we can bound the whole product $\delta_f(s)\langle s,r\rangle \leq |\delta_f(s)|\langle s,r\rangle \leq L\|x-y\|_2\langle s,r\rangle$. Then, this has to be integrated over the half-sphere $s \in S^d$, $\langle s,r\rangle \geq 0$, which we denote by $\mathcal{S}^d_+$. Consequently, using (B.15) we get

$$\|\nabla f_\nu(x) - \nabla f_\nu(y)\|_2 = \frac{d}{\nu}\mathbb{E}_s\delta_f(s)\langle s,r\rangle \leq \frac{d}{\nu}L\|x-y\|_2\mathbb{E}_{s\in\mathcal{S}^d_+}\langle s,r\rangle. \tag{B.16}$$

Note that the expectation over the half sphere $s \sim \mathcal{U}(\mathcal{S}^d_+)$ of projection of s onto the one direction r is

$$\mathbb{E}_{s\in\mathcal{S}^d_+}\langle s,r\rangle = \frac{2}{Vol(\mathcal{S}^d)}\int_{\theta\in[0,\pi/2]} Vol(\mathcal{S}^{d-1})\sin\theta(\cos\theta)^{d-1}d\theta = \frac{Vol(\mathcal{S}^{d-1})}{Vol(\mathcal{S}^d)}\int_0^1 t^{d-1}dt. \tag{B.17}$$

In the above $Vol(\mathcal{S}^d)$ denotes the surface area of $\mathcal{S}^d$. Then, we can use the following well known relations.
If d is even:

$$Vol(\mathcal{S}^d) = \frac{(2\pi)^{d/2}}{(d-2)!!},\ Vol(\mathcal{S}^{d-1}) = \frac{2(2\pi)^{(d-2)/2}}{(d-3)!!}.$$

If d is odd:

$$Vol(\mathcal{S}^d) = \frac{2(2\pi)^{(d-1)/2}}{(d-2)!!},\ Vol(\mathcal{S}^{d-1}) = \frac{(2\pi)^{(d-1)/2}}{(d-3)!!}.$$

Therefore, if d is even: $\frac{Vol(\mathcal{S}^{d-1})}{Vol(\mathcal{S}^d)} = \frac{(d-2)!!}{\pi(d-3)!!} \leq \sqrt{d}$. If d is odd: $\frac{Vol(\mathcal{S}^{d-1})}{Vol(\mathcal{S}^d)} = \frac{(d-2)!!}{2(d-3)!!} \leq \sqrt{d}$. Hence, from (B.17) we get

$$\mathbb{E}_{s\in\mathcal{S}^d_+}\langle s,r\rangle = \frac{Vol(\mathcal{S}^{d-1})}{Vol(\mathcal{S}^d)}\frac{1}{d} \leq \frac{\sqrt{d}}{d} \leq \frac{1}{\sqrt{d}}.$$

Finally, from (B.16) and we can conclude the statement of the property:

$$\|\nabla f_\nu(x) - \nabla f_\nu(y)\|_2 \leq \frac{d}{\nu\sqrt{d}}L\|x-y\|_2 = \frac{\sqrt{d}L}{\nu}\|x-y\|_2.$$

■

B.9 Proof of property 3 in Fact 5

By definition we have $|f_\nu(x) - f_\nu(y)| = \left|\mathbb{E}_{b\sim U(\mathbb{B})}\big(f(x+\nu b) - f(y+\nu b)\big)\right|$. Then, using Jensen's inequality for $|\cdot|$, we can swap $\mathbb{E}_b$ and the absolute value $|\cdot|$ in the above, and obtain:

$$|f_\nu(x) - f_\nu(y)| \leq \mathbb{E}_{b\sim U(\mathbb{B})}|f(x+\nu b) - f(y+\nu b)| \leq \mathbb{E}_{b\sim U(\mathbb{B})} L\|x-y\| = L\|x-y\|.$$

From the above, any directional derivative is bounded by L:

$$\frac{\langle \nabla f_\nu(x), u\rangle}{\|u\|} = \lim_{t\to 0} \frac{\langle \nabla f_\nu(x), tu\rangle}{\|tu\|} = \lim_{t\to 0} \frac{|f_\nu(x+tu) - f_\nu(x)|}{\|tu\|} \leq L \quad \forall x, u \in \mathbb{R}^d.$$

Consequently, the norm of the gradient $\nabla f_\nu(x)$ is bounded by L:

$$\|\nabla f_\nu(x)\| \leq L \quad \forall x \in \mathbb{R}^d.$$

B.10 Proof of Lemma 11

The first part follows from the definition of the smoothed approximation f_ν^i, the fact that the noise is zero-mean, and Lemma 9. The second part follows from the smoothness.

Proof. Note that

$$|f^i(x+\nu b_j) - f^i(x)| \leq L_i \nu \|b_i\| \leq L_i \nu.$$

Hence, also using the Fact 5, we have

$$|f^i(x+\nu b_j) - f_\nu^i(x)| \leq |f^i(x+\nu b_j) - f^i(x)| + |f^i(x) - f_\nu^i(x)| \leq 2L_i\nu.$$

Therefore, finally we can show the following bound on the variance of $F_{\nu,n}^i(x,\xi)$:

$$\begin{aligned}
\mathbb{E}\left(\sum_{j=1}^n \frac{F^i(x+\nu b_j)}{n} - f_\nu^i(x)\right)^2 &= \mathbb{E}\left(\frac{f^i(x+\nu b_j) + \xi_j}{n} - f_\nu^i(x)\right)^2 \\
&\leq \mathbb{E}\left(\sum_{j=1}^n \frac{(f^i(x+\nu b_j) - f_\nu^i(x))}{n}\right)^2 + \mathbb{E}\left(\frac{\sum_{j=1}^n \xi_j}{n}\right)^2 \\
&\leq \frac{4L_i^2\nu^2}{n} + \frac{\sigma_i^2}{n}.
\end{aligned}$$

■

Next, we prove the third part:

Proof. The deviation of the gradient estimators

$$G^j(x_k,\nu) - \nabla f^j_\nu(x_k) = \frac{1}{n}\sum_{l=1}^{n}\left[\underbrace{\left(d\frac{f^i(x_t+\nu s_{tj}) - f^i(x_t)}{\nu}s_{tj} - \nabla f^j_\nu(x_t)\right)}_{v^i_j} + \underbrace{d\frac{\xi^{i+}_{tj} - \xi^{i-}_{tj}}{\nu}s_{tj}}_{u^j_l}\right]. \tag{B.18}$$

Then,

$$\begin{aligned}\mathbb{E}\|G^j(x_t,\nu) - \nabla f^j_\nu(x_t)\|^2 &= \mathbb{E}\|\frac{1}{n}\sum_{l=1}^{n} v^j_l\|^2 + \mathbb{E}\|\frac{1}{n}\sum_{l=1}^{n_k} u^j_l\|^2 + 2\mathbb{E}\langle\frac{1}{n}\sum_{l=1}^{n_k} v^j_l, \frac{1}{n}\sum_{l=1}^{n_k} u^j_l\rangle\\ &= \mathbb{E}\|\frac{1}{n}\sum_{l=1}^{n} v^j_l\|^2 + \mathbb{E}\|\frac{1}{n}\sum_{l=1}^{n} u^j_l\|^2,\end{aligned} \tag{B.19}$$

since the noise ξ^i_j is zero-mean, and independent on v^i_j. Next, we are going to bound each of the terms in the summand above. From Stokes' theorem [FKM05] we know that $\mathbb{E}v^j_l = 0$. Using L_i-Lipschitzness of $f^i(x)$ for $i \in \{0,\dots,m\}$ we can derive:

$$\|v^j_l\| = \left\|d\frac{f^j(x_k+\nu s_{kl}) - f^j(x_k)}{\nu}s_{kl} - \nabla f^j_\nu(x_k)\right\| \leq (d+1)L. \tag{B.20}$$

Since $\{v^j_l\}_{l=1,\dots,n_k}$ are i.i.d. zero-mean variables, we have

$$\mathbb{E}\left\|\frac{1}{n}\sum_{l=1}^{n_k} v^j_l\right\|^2 = \frac{1}{n}\sum_{l=1}^{n}\mathbb{E}\|v^j_l\|^2 = \frac{1}{n}(d+1)^2L_i^2 \tag{B.21}$$

Also, we can bound: $\mathbb{E}\|\frac{1}{n}\sum_{l=1}^{n_k} u^j_l\|^2 \leq \frac{1}{n}\frac{2d^2\sigma_i^2}{\nu^2}$. Combining the above with Eq. (B.19), we get:

$$\mathbb{E}\|G^j(x_t,\nu) - \nabla f^j_\nu(x_t)\|^2 \leq \frac{1}{n_k}\left((d+1)^2L^2 + \frac{2d^2\sigma^2}{\nu^2}\right) \leq \frac{(d+1)^2}{n}\left(L_i^2 + \frac{2\sigma_i^2}{\nu^2}\right). \tag{B.22}$$

■

APPENDIX C

Additional materials for Chapter 5

C.1 Internal model

The graphical probabilistic model of POMDP is shown at Figure C.1, where the state consists of two parts $s_t = [d_t, z_t]$. Here, d_t denotes the deterministic part that is determined only by dynamics and previous action and state pair $d_t \sim P_d(\cdot|a_{t-1}, d_{t-1}, z_{t-1})$ where p_d is a Dirac distribution (denoted by green arrows). Whereas z_t determines the stochastic part, which is a random variable dependent on d_t only, i.e., $z_t \sim P_z(\cdot|d_t)$ (denoted by blue arrow). And finally, o_t is the random variable corresponding to observation whose distribution depends on the state $[d_t, z_t]$, that is, $o_t \sim P_o(\cdot|d_t, z_t)$ (denoted by orange arrows).

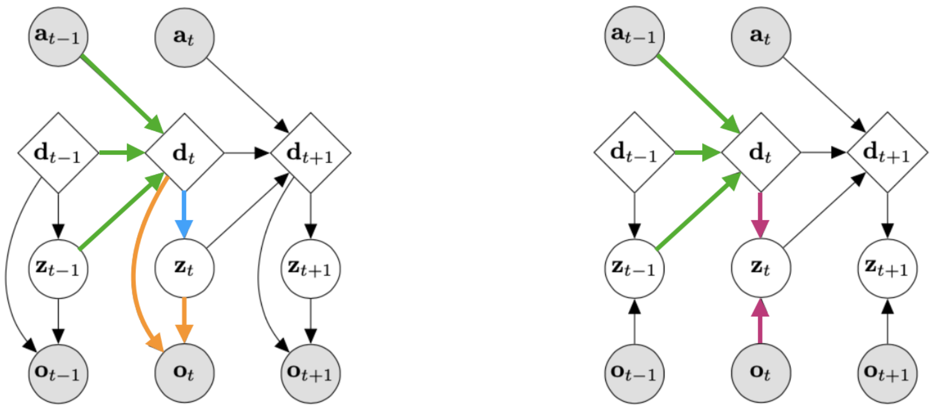


Figure C.1: (a) Probabilistic graphical model of the POMDP; (b) Probabilistic graphical model of the posterior inference.

We infer the transition density from observations using the Recurrent State Space Model (RSSM) introduced in Hafner, Lillicrap, Fischer, Villegas, Ha, Lee, and Davidson [Haf+19a]. Our RSSM network consists of two parts, one is responsible for modelling the POMDP: RSSM$_\theta$, and the second one is responsible for inferring the posterior distribution after getting the observations RSSM$_\phi$, and is used for training the network. The probabilistic distributions P_z and P_o it models using Gaussian distributions with mean and variance predicted by NNs: $P_o(\cdot|d_t, z_t) = P_{\theta_1}(\cdot|d_t, z_t) = \mathcal{N}(\mu_{\theta_1}, \Sigma_{\theta_1})$ and $P_z(\cdot|d_t) =$

$P_{\theta_2}(\cdot|d_t) = \mathcal{N}(\mu_{\theta_2}, \Sigma_{\theta_2})$. The deterministic part of the dynamics is modeled by NN directly: $P_d(d_{t-1}, z_{t-1}, a_{t-1}) = \delta(f_{\theta_3}(d_{t-1}, z_{t-1}, a_{t-1}))$. The parameter θ consists of $\theta = \{\theta_1, \theta_2, \theta_3\}$. The second part of RSSM$_\phi$ learns the posterior distribution $z_t \sim \hat{P}_\phi(\cdot|o_t, d_t) = \mathcal{N}(\mu_\phi, \Sigma_\phi)$.

C.2 Training the internal model

In more details, our network, parametrized by parameters $\theta = \{\theta_1, \theta_2, \theta_3\}$, and ϕ, allows to model the generative process of POMDP which is defined by the joint probability:

$$P_\theta(o_{1:T}, z_{1:T}, d_{1:T}|a_{0:T-1}, z_0, d_0) = \prod_{t=1}^{T} P_{\theta_1}(o_t|z_t, d_t) P_{\theta_2}(z_t|d_t) P_{\theta_3}(d_t|z_{t-1}, d_{t-1}, a_{t-1}).$$

We train the network such that the last term in the above equation $P_{\theta_3}(d_t|z_{t-1}, d_{t-1}, a_{t-1}) = \delta(d_t - f_{\theta_3}(z_{t-1}, d_{t-1}, a_{t-1}))$, where $f_{\theta_3}(z_{t-1}, d_{t-1}, a_{t-1})$ is the deterministic recurrent function of the network, and $\delta(\cdot)$ is the Dirac-function. That is, $d_t = f_{\theta_3}(z_{t-1}, d_{t-1}, a_{t-1})$. The density $P_{\theta_2}(z_t|d_t)$ is a Gaussian distribution whose mean and variance are the outputs of a neural network. Finally, $P_{\theta_3}(o_t|z_t, d_t)$ expresses the observation reconstruction from the latent state s_t. The second part of the network is responsible for modelling the posterior:

$$P_\phi(z_{1:T}, d_{1:T}|o_{1:T}, a_{0:T-1}, z_0, d_0) = \prod_{t=1}^{T} \hat{P}_\phi(z_t|o_t, d_t) P_{\theta_3}(d_t|z_{t-1}, d_{t-1}, a_{t-1}).$$

For training the above network parameters $\theta = \{\theta_1, \theta_2, \theta_3\}, \phi$, we use the gradient steps on the following loss function:

$$\mathcal{L}(\theta, \phi, o_{1:T}) = \sum_{t=1}^{T} \mathbb{E}_{\hat{P}_\phi(z_{1:T}|o_{1:T}, a_{0:T-1}, z_0, d_0)} \left[KL(\hat{P}_\phi(z_t|o_t, d_t) || P_{\theta_2}(z_t|d_t)) - \log P_{\theta_1}(o_t|z_t, d_t) \right],$$

where KL denotes the Kullback-Leibler Divergence.

C.3 Learning critics

For the task and safety critics, we use the reward and the cost value functions. $\mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r_\tau | s_t\right]$ which is given as a dense neural network with parameters ψ and discount factor γ. Similarly, we model the cost value by $v^\pi_{\psi_c}(\mathbf{s}_\tau) \approx \mathbb{E}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} c_\tau | s_t\right]$ with parameter vector ψ_c. The policy and value models are trained cooperatively as typical in policy iteration: the action model aims to maximize an estimate of the value, while the value model aims to match an estimate of the value that changes as the policy model changes. As in Hafner, Lillicrap, Ba, and Norouzi [Haf+19b], we use TD(λ) [SB18] to trade-off the bias and variance of the critics with bootstrapping and Monte-Carlo value estimation. We denote $\mathbf{V}_\lambda(s_\tau)$ as the TD(λ) value as presented in Hafner, Lillicrap, Ba, and Norouzi [Haf+19b], *that is defined recursively using* $v^{\pi}_{\psi}(s_\tau)$. In the same way, we define $\mathbf{V}_{\lambda,c}(\mathbf{s}_\tau)$ as the TD(λ) value of the constraint. The models $v^{\pi}_{\psi}, v^{\pi}_{\psi_c}$ at a turn are updated via minimizing the regression loss over ϕ, ψ_c, for the fixed policy π and dynamics p_θ:

$$\mathcal{L}_{v^{\pi}_{\psi}}(\psi) = \mathbb{E}_{\mathbf{a}_t\sim\pi,\mathbf{s}_{\tau:\tau+H}\sim p_\theta}\left[\frac{1}{2}\sum_{t=\tau}^{\tau+H}\left(v^{\pi}_{\psi}(\mathbf{s}_t) - \mathbf{V}_{\lambda}(\mathbf{s}_t)\right)^2\right]. \tag{C.1}$$

$$\mathcal{L}_{v^{\pi}_{\psi_c}}(\psi_c) = \mathbb{E}_{\mathbf{a}_t\sim\pi,\mathbf{s}_{\tau:\tau+H}\sim p_\theta}\left[\frac{1}{2}\sum_{t=\tau}^{\tau+H}\left(v^{\pi}_{\psi_c}(\mathbf{s}_t) - \mathbf{V}_{\lambda,c}(\mathbf{s}_t)\right)^2\right]. \tag{C.2}$$

We denote $\mathbf{V}_{\lambda}(s_\tau)$ as the TD(λ) value as presented in Hafner, Lillicrap, Ba, and Norouzi [Haf+19b].

$$\mathbf{V}_R(s_\tau) := \mathbb{E}_{\pi_\xi,p_\theta}\left(\sum_{n=\tau}^{\tau+H} r_n\right) \tag{C.3}$$

$$\mathbf{V}^k_N(s_\tau) := \mathbb{E}_{\pi_\xi,p_\theta}\left(\sum_{n=1}^{H-1} \gamma^{n-\tau} r_n + \gamma^{h-\tau} v^{\pi}_{\psi}(s_\tau)\right), \text{ with } h = \min(\tau+k, t+H) \tag{C.4}$$

$$\mathbf{V}_{\lambda}(s_\tau) := (1-\lambda)\sum_{n=1}^{H-1} \lambda^{n-1}\mathbf{V}^n_N(s_\tau) + \lambda^{H-1}\mathbf{V}^H_N(s_\tau), \tag{C.5}$$

where the expectations are estimated under the imagined trajectories.

Bibliography

[AHR12] J. D. Abernethy, E. Hazan, and A. Rakhlin. “Interior-point methods for full-information and bandit online learning”. In: *IEEE Transactions on Information Theory* 58.7 (2012), pp. 4164–4175.

[Ach+17] J. Achiam, D. Held, A. Tamar, and P. Abbeel. *Constrained Policy Optimization*. 2017. arXiv: `1705.10528 [cs.LG]`.

[AZL20] S. Akhauri, L. Zheng, and M. Lin. *Enhanced Transfer Learning for Autonomous Driving with Systematic Accident Simulation*. 2020. URL: `https://arxiv.org/abs/2007.12148`.

[Alt99] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

[AAT19] S. Amani, M. Alizadeh, and C. Thrampoulidis. “Linear stochastic bandits under safety constraints”. In: *arXiv preprint arXiv:1908.05814* (2019).

[Arj+19] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth. “Lower bounds for non-convex stochastic optimization”. In: *arXiv preprint arXiv:1912.02365* (2019).

[As+22] Y. As, I. Usmanova, S. Curi, and A. Krause. “Constrained Policy Optimization via Bayesian World Models”. In: *ArXiv* (2022). URL: `https://arxiv.org/abs/2201.09802`.

[BP16] F. Bach and V. Perchet. “Highly-smooth zero-th order online optimization”. In: *Conference on Learning Theory*. 2016, pp. 257–283.

[BG18] K. Balasubramanian and S. Ghadimi. “Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 3455–3464.

[BEN09] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[BN98] A. Ben-Tal and A. Nemirovski. “Robust convex optimization”. In: *Mathematics of operations research* 23.4 (1998), pp. 769–805.

[BN99] A. Ben-Tal and A. Nemirovski. “Robust solutions of uncertain linear programs”. In: *Operations research letters* 25.1 (1999), pp. 1–13.

[BN00] A. Ben-Tal and A. Nemirovski. “Robust solutions of linear programming problems contaminated with uncertain data”. In: *Mathematical programming* 88.3 (2000), pp. 411–424.

[Ber+21] A. Berahas, L. Cao, K. Choromanski, and K. Scheinberg. “A Theoretical and Empirical Comparison of Gradient Approximations in Derivative-Free Optimization”. In: *Foundations of Computational Mathematics* (May 2021).

[BKS16] F. Berkenkamp, A. Krause, and A. P. Schoellig. “Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics”. In: *arXiv preprint arXiv:1602.04450* (2016).

[Ber+17] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause. *Safe Model-based Reinforcement Learning with Stability Guarantees*. 2017. arXiv: 1705.08551 [stat.ML].

[BBC11] D. Bertsimas, D. B. Brown, and C. Caramanis. “Theory and applications of robust optimization”. In: *SIAM review* 53.3 (2011), pp. 464–501.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BLE17] S. Bubeck, Y. T. Lee, and R. Eldan. “Kernel-based methods for bandit convex optimization”. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. 2017, pp. 72–85.

[CKK+96] A. R. Cassandra, L. P. Kaelbling, J. Kurien, et al. “Acting under uncertainty: discrete Bayesian models for mobile-robot navigation.” In: *IROS*. Vol. 96. 1996, pp. 963–972.

[CZK19] L. Chen, M. Zhang, and A. Karbasi. “Projection-free bandit convex optimization”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 2047–2056.

[Cho+15] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. “Risk-Constrained Reinforcement Learning with Percentile Risk Criteria”. In: *CoRR* abs/1512.01629 (2015). arXiv: 1512.01629. URL: http://arxiv.org/abs/1512.01629.

[CUH15] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. 2015. URL: https://arxiv.org/abs/1511.07289.

[DHK08] V. Dani, T. P. Hayes, and S. M. Kakade. “Stochastic linear optimization under bandit feedback”. In: (2008).

[DR13] E. Díaz-Francés and F. Rubio. “On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables”. In: *Statistical Papers* 54.2 (May 2013), pp. 309–323. URL: https://ideas.repec.org/a/spr/stpapr/v54y2013i2p309-323.html.

[DS14] N. R. Draper and H. Smith. *Applied regression analysis*. Vol. 326. John Wiley & Sons, 2014.

[DR18] F. Duarte and C. Ratti. “The impact of autonomous vehicles on cities: A review”. In: *Journal of Urban Technology* 25.4 (2018), pp. 3–18.

[Duc+15] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. “Optimal Rates for Zero-Order Convex Optimization: The Power of Two Function Evaluations”. In: *IEEE Transactions on Information Theory* 61.5 (2015), pp. 2788–2806.

[DSS21] P. Dvurechensky, S. Shtern, and M. Staudigl. “First-Order Methods for Convex Optimization”. In: *EURO Journal on Computational Optimization* 9 (2021), p. 100015. URL: https://www.sciencedirect.com/science/article/pii/S2192440621001428.

[Fai+19] A. Faisal, M. Kamruzzaman, T. Yigitcanlar, and G. Currie. “Understanding autonomous vehicles”. In: *Journal of transport and land use* 12.1 (2019), pp. 45–72.

[Faz+19] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas. “Efficient and accurate estimation of lipschitz constants for deep neural networks”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[Fer+20] M. Fereydounian, Z. Shen, A. Mokhtari, A. Karbasi, and H. Hassani. “Safe Learning under Uncertain Objectives and Constraints”. In: *arXiv preprint arXiv:2006.13326* (2020).

[FKM05] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. “Online convex optimization in the bandit setting: gradient descent without a gradient”. In: *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2005, pp. 385–394.

[FW56] M. Frank and P. Wolfe. “An algorithm for quadratic programming”. In: *Naval Research Logistics (NRL)* 3.1-2 (1956), pp. 95–110.

[FG16] R. M. Freund and P. Grigas. “New analysis and results for the Frank–Wolfe method”. In: *Mathematical Programming* 155.1-2 (2016), pp. 199–230.

[GK20] D. Garber and B. Kretzu. “Improved regret bounds for projection-free bandit convex optimization”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2196–2206.

[Gar+14] J. Gardner, M. Kusner, Zhixiang, K. Weinberger, and J. Cunningham. “Bayesian Optimization with Inequality Constraints”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 937–945. URL: https://proceedings.mlr.press/v32/gardner14.html.

[GL13] S. Ghadimi and G. Lan. “Stochastic first-and zeroth-order methods for nonconvex stochastic programming”. In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2341–2368.

[Haf+19a] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. “Learning Latent Dynamics for Planning from Pixels”. In: *International Conference on Machine Learning*. 2019, pp. 2555–2565.

[Haf+19b] D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. “Dream to Control: Learning Behaviors by Latent Imagination”. In: *CoRR* abs/1912.01603 (2019). arXiv: 1912.01603. URL: http://arxiv.org/abs/1912.01603.

[HO01] N. Hansen and A. Ostermeier. “Completely Derandomized Self-Adaptation in Evolution Strategies.” In: *Evol. Comput.* 9.2 (2001), pp. 159–195. URL: http://dblp.uni-trier.de/db/journals/ec/ec9.html#HansenO01.

[HL16] E. Hazan and H. Luo. “Variance-reduced and projection-free stochastic optimization”. In: *International Conference on Machine Learning*. 2016, pp. 1263–1271.

[HY18] O. Hinder and Y. Ye. “A one-phase interior point method for nonconvex optimization”. In: *arXiv preprint arXiv:1801.03072* (2018).

[HY19] O. Hinder and Y. Ye. “A polynomial time log barrier method for problems with nonconvex constraints”. In: *arXiv preprint: https://arxiv.org/pdf/1807.00404.pdf* (2019).

[INF18] D. Isele, A. Nakhaei, and K. Fujimura. “Safe reinforcement learning on autonomous vehicles”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–6.

[Jag13] M. Jaggi. “Revisiting Frank-Wolfe: projection-free sparse convex optimization”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28*. JMLR. org. 2013, pp. I–427.

[JHA15] R. Jenatton, J. Huang, and C. Archambeau. “Adaptive algorithms for online convex optimization with long-term constraints”. In: *arXiv preprint arXiv:1512.07422* (2015).

[Jud+13] A. B. Juditsky, G. Lan, A. S. Nemirovski, and A. Shapiro. *Stochastic Approximation approach to Stochastic Programming*. Research Report. http://www.optimization-online.org/DB_HTML/2007/09/1787.html. LJK, 2013. URL: https://hal.archives-ouvertes.fr/hal-00853911.

[Kat+15] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. “An open approach to autonomous vehicles”. In: *IEEE Micro* 35.6 (2015), pp. 60–68.

[KE95] J. Kennedy and R. C. Eberhart. “Particle swarm optimization”. In: *Proceedings of the IEEE International Conference on Neural Networks*. 1995, pp. 1942–1948.

[Kir+19] J. Kirschner, M. Mutny, N. Hiller, R. Ischebeck, and A. Krause. “Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces”. In: *arXiv preprint arXiv:1902.03229* (2019).

[KS96] S. Koenig and R. G. Simmons. “Unsupervised learning of probabilistic models for robot navigation”. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*. Vol. 3. IEEE. 1996, pp. 2301–2308.

[Kol+18] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause. “Learning-based model predictive control for safe exploration”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE. 2018, pp. 6059–6066.

[KT51] H. Kuhn and A. Tucker. “pp. 481–492 in: Nonlinear Programming”. In: *Proc. 2nd Berkeley Symp. Math. Stat. Prob.(J. Neyman, ed.), Univ. of Calif. Press, Berkeley, CA*. Vol. 14. 1951, p. 208.

[LJ13] S. Lacoste-Julien and M. Jaggi. “An affine invariant linear convergence analysis for Frank-Wolfe algorithms”. In: *arXiv preprint arXiv:1312.7864* (2013).

[Lan20] G. Lan. *First-order and Stochastic Optimization Methods for Machine Learning*. Springer Series in the Data Sciences. Springer International Publishing, 2020. URL: https://books.google.ch/books?id=7dTkDwAAQBAJ.

[Lan13] G. Lan. “The complexity of large-scale convex programming under a linear optimization oracle”. In: *arXiv preprint arXiv:1309.5550* (2013).

[LLG04] M. Luersen, R. Le Riche, and F. A. Guyon. “Constrained, globalized, and bounded Nelder–Mead method for engineering optimization.” In: *Struct Multidisc Optim 27* (2004), pp. 43–54.

[MJY12] M. Mahdavi, R. Jin, and T. Yang. “Trading regret for efficiency: online convex optimization with long term constraints”. In: *Journal of Machine Learning Research* 13.Sep (2012), pp. 2503–2528.

[Mai+18] M. Maier, A. Rupenyan, R. Zwicker, M. Akbari, and K. Wegener. “Turning: Autonomous Process Set-up through Bayesian Optimization and Gaussian Process Models”. In: *12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gulf of Naples, Italy* (2018).

[Mak+21] A. Makarova, I. Usmanova, I. Bogunovic, and A. Krause. “Risk-averse Heteroscedastic Bayesian Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021. URL: https://openreview.net/forum?id=QO93ev_yPqn.

[MF67] O. Mangasarian and S. Fromovitz. “The Fritz John necessary optimality conditions in the presence of equality and inequality constraints”. In: *Journal of Mathematical Analysis and Applications* 17.1 (1967), pp. 37–47. URL: https://www.sciencedirect.com/science/article/pii/0022247X67901631.

[Moh+20] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. *Monte Carlo Gradient Estimation in Machine Learning*. 2020. arXiv: 1906.10652 [stat.ML].

[NM65] J. A. Nelder and R. Mead. “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. eprint: https://academic.oup.com/comjnl/article-pdf/7/4/308/1013182/7-4-308.pdf. URL: https://doi.org/10.1093/comjnl/7.4.308.

[Nem+09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. "Robust Stochastic Approximation Approach to Stochastic Programming". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1574–1609. eprint: https://doi.org/10.1137/070704277. URL: https://doi.org/10.1137/070704277.

[NY85] A. S. Nemirovsky and D. B. Yudin. "Problem Complexity and Method Efficiency in Optimization". In: *SIAM Review* 27.2 (1985), pp. 264–265. eprint: https://doi.org/10.1137/1027074. URL: https://doi.org/10.1137/1027074.

[NW06] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[NM10] K. Nolde and M. Morari. "Electrical load tracking scheduling of a steel plant". In: *Computers & Chemical Engineering* 34.11 (2010), pp. 1899–1903. URL: https://www.sciencedirect.com/science/article/pii/S0098135410000244.

[Now+19] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganiere, and J. Rebut. *How much real data do we actually need: Analyzing object detection performance using synthetic and real data*. 2019. URL: https://arxiv.org/abs/1907.07061.

[Pow69] M. J. D. Powell. "A method for nonlinear constraints in minimization problems". In: 1969.

[Pri19] C. Price. "A modified Nelder-Mead barrier method for constrained optimization". In: *Numerical Algebra, Control, and Optimization* 11 (Jan. 2019).

[RW05] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[Rat+21] L. Rattunde, I. Laptev, E. D. Klenske, and H.-C. Möhring. "Safe optimization for feedrate scheduling of power-constrained milling processes by using Gaussian processes". In: *Procedia CIRP* 99 (2021). 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, pp. 127–132. URL: https://www.sciencedirect.com/science/article/pii/S2212827121002821.

[RAA19] A. Ray, J. Achiam, and D. Amodei. "Benchmarking Safe Exploration in Deep Reinforcement Learning". In: (2019).

[Rec89] I. Rechenberg. "Evolution Strategy: Nature's Way of Optimization". In: *Optimization: Methods and Applications, Possibilities and Limitations*. Ed. by H. W. Bergmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 106–126.

[RM51] H. Robbins and S. Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. URL: https://doi.org/10.1214/aoms/1177729586.

[SAR18] W. Schwarting, J. Alonso-Mora, and D. Rus. “Planning and decision-making for autonomous vehicles”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (2018), pp. 187–210.

[Sha13] O. Shamir. “On the complexity of bandit and derivative-free stochastic convex optimization”. In: *Conference on Learning Theory*. 2013, pp. 3–24.

[SP97] R. Storn and K. V. Price. “Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.” In: *J. Glob. Optim.* 11.4 (1997), pp. 341–359. URL: http://dblp.uni-trier.de/db/journals/jgo/jgo11.html#StornP97.

[Sui+15a] Y. Sui, A. Gotovos, J. Burdick, and A. Krause. “Safe Exploration for Optimization with Gaussian Processes”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 997–1005. URL: http://proceedings.mlr.press/v37/sui15.html.

[Sui+15b] Y. Sui, A. Gotovos, J. Burdick, and A. Krause. “Safe exploration for optimization with Gaussian processes”. In: *International Conference on Machine Learning*. 2015, pp. 997–1005.

[SDK17] W. Sun, D. Dey, and A. Kapoor. “Safety-Aware Algorithms for Adversarial Contextual Bandit”. In: *International Conference on Machine Learning*. 2017, pp. 3280–3288.

[SB18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[Usm+22] I. Usmanova, Y. As, M. Kamgarpour, and A. Krause. *Log Barriers for Safe Black-box Optimization with Application to Safe Reinforcement Learning*. 2022. URL: https://arxiv.org/abs/2207.10415.

[Usm+21] I. Usmanova, M. Kamgarpour, A. Krause, and K. Levy. “Fast Projection Onto Convex Smooth Constraints”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 10476–10486. URL: https://proceedings.mlr.press/v139/usmanova21a.html.

[UKK19] I. Usmanova, A. Krause, and M. Kamgarpour. “Safe Convex Learning under Uncertain Constraints”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 2106–2114.

[UKK20] I. Usmanova, A. Krause, and M. Kamgarpour. “Safe non-smooth black-box optimization with application to policy search”. In: *Learning for Dynamics and Control*. 2020, pp. 980–989.

[VDB21] S. Vaswani, B. Dubois-Taine, and R. Babanezhad. “Towards Noise-adaptive, Problem-adaptive Stochastic Gradient Descent”. working paper or preprint. Nov. 2021. URL: https://hal.archives-ouvertes.fr/hal-03456663.

[WZ21] K. P. Wabersich and M. N. Zeilinger. *A predictive safety filter for learning-based control of constrained nonlinear dynamical systems*. 2021. arXiv: 1812.05506 [cs.SY].

[Wen+20] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng. "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–7.

[YNS10] F. Yousefian, A. Nedić, and U. V. Shanbhag. "Convex nondifferentiable stochastic optimization: A local randomized smoothing technique". In: *Proceedings of the 2010 American Control Conference*. IEEE. 2010, pp. 4875–4880.

[YNW17] H. Yu, M. Neely, and X. Wei. "Online Convex Optimization with Stochastic Constraints". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1427–1437.

[YN16] H. Yu and M. J. Neely. "A Low Complexity Algorithm with $O(\sqrt{T})$ Regret and Finite Constraint Violations for Online Convex Optimization with Long Term Constraints". In: *arXiv preprint arXiv:1604.02218* (2016).

[Zha+21] S. Zhang, L. Wen, H. Peng, and H. E. Tseng. "Quick learner automated vehicle adapting its roadmanship to varying traffic cultures with meta reinforcement learning". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2021, pp. 1745–1752.

[ZG17] H. Zhao and G. J. Gordon. "Frank-Wolfe Optimization for Symmetric-NMF under Simplicial Constraint". In: *CoRR* abs/1706.06348 (2017). arXiv: 1706.06348. URL: http://arxiv.org/abs/1706.06348.

[ZQW20] W. Zhao, J. P. Queralta, and T. Westerlund. "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey". In: *CoRR* abs/2009.13303 (2020). arXiv: 2009.13303. URL: https://arxiv.org/abs/2009.13303.