THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présentée et soutenue le *13/11/2023* **par :**

Etienne de Montbrun

Théorie des Jeux et Apprentissage Séquentiel : Montée-descente de gradient, Optimisation, Approximation et Certification

---

JURY

Aude Rondepierre	INSA Toulouse	Président du Jury
Panayotis Mertikopoulos	Université Grenoble Alpes	Rapporteur
Wouter Koolen	University of Twente	Rapporteur
Jérôme Renault	Toulouse School of Economics	Directeur de Thèse
Sébastien Gerchinovitz	IRT Saint Exupéry	Directeur de Thèse
Vianney Perchet	CREST, ENSAE	Membre du Jury
Sylvain Sorin	Sorbonne Université	Membre du Jury

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Toulouse School of Economics – Research (UMR TSE-R)
Directeur(s) de Thèse :
Jérôme Renault et *Sébastien Gerchinovitz*
Rapporteurs :
Panayotis Mertikopoulos et *Wouter Koolen*

Abstract

Théorie des Jeux et Apprentissage Séquentiel : Montée-descente de gradient, Optimisation, Approximation et Certification

Résumé : Cette thèse se situe dans les domaines de la théorie des jeux et de l'apprentissage séquentiel, en abordant trois problèmes distincts par l'étude de trois algorithmes optimistes différents. Tout d'abord, nous approfondissons l'étude d'OGDA, une variante de l'algorithme de montée-descente de gradient, dans le cadre des jeux bilinéaires sans contraintes. OGDA améliore ses performances grâce à un terme de gradient supplémentaire dans la mise à jour. Notre deuxième domaine de recherche est l'optimisation multi-fidélité pour les fonctions lipschitziennes, où nous nous intéressons plus particulièrement à la notion de certificat. Nous introduisons c.MF-DOO, un algorithme certifié s'appuyant sur la maximisation de bornes supérieures de la fonction à optimiser. Enfin, nous examinons GreedyBox, un algorithme qui minimise les erreurs d'approximation L^p des fonctions monotones grâce à une approche dite "gourmande".
Notre étude examine les dynamiques, les vitesses de convergence, les limites ainsi que certaines améliorations de ces trois algorithmes, et conduit à de nouvelles perspectives sur les avantages de la certification et de l'optimisme.

Mots-clés : Théorie de jeux, optimisation multi-fidélité, optimisation globale, algorithmes de bandits, algorithmes séquentiels, Montée-descente de Gradient Optimiste, approximation L^p, intégration numérique.

Game Theory and Online Learning: Gradient Descent Ascent, Optimization, Approximation and Certification

Abstract: This thesis delves into the topics of game theory and online learning, addressing three distinct problems, with the study of three different optimistic algorithms. Firstly, we strengthen the study of OGDA, a variant of the gradient descent ascent algorithm, in the framework of unconstrained bilinear games. It improves its performance thanks to an extra gradient update at each time step. Our second focus is multi-fidelity zeroth-order optimization of Lipschitz functions, where we emphasize on certification. We introduce c.MF-DOO, a certified algorithm built on the maximization of a surrogate function. Finally, we consider GreedyBox, which minimizes L^p approximation errors of monotone functions through a greedy approach.
Our study scrutinizes the mechanics, convergence rates, limitations and some improvements of these algorithms, and leads to new insights on the benefit of certification and optimism.

Keywords: Game theory, multi-fidelity optimization, global optimization, bandits algorithms, sequential algorithms, Optimistic Gradient Descent Ascent, L^p-approximation, numerical integration.

Acknowledgments

Let no one ever come to you without
leaving better and happier

Mother Teresa

First, I want to thank all the people who left me better than when I first met them, and who helped through these three years of PhD. Above all I would like to thank Sébastien and Jérôme, who made this scientific adventure possible. They kindly taught me everything I needed for this PhD, from the bibliography of the first months, through the core of research and its difficulty, to many tips for redaction (for which I am, unfortunately, a very slow learner). They were of great support during the hardest periods, when every track led to dead ends, it is their optimism that helped me to hold on. Thank you very much for the time you spent for me albeit your full timetable.
I want to thank Panayotis and Wouter. Thanks to them and their readiness, finding reviewers was never that simple. Thank you very much for the time you took to read this manuscript. I additionally thank Panayotis for the organization of the Alpine Game Theory Symposium that I greatly appreciated.
I also thank the members of the jury: with the previously quoted, I thank Aude Rondepierre, Vianney Perchet and Sylvain Sorin.

I am also deeply indebted to all the people I worked with. With the one already cited, I want to add Pierre Gaillard, for the two intensive weeks at Grenoble, and for the hospitality of the INRIA. I now want to thank all those who I had the pleasure to work with. I deeply thank Maurizio, Lukas and Thi, Tâm, Pablo, Ryan, Joseph, and Amadou at TSE. Thank you for sharing an office with me and for the numerous lunches with great discussions.
Thank you Léo, Louis, Agustin, Thibaut, Tom, Fanny, Franck, Justin, the Davids, Victor and Estelle and all the other members of DEEL and Confiance that made these three PhD years fly by so fast. Thank you for the chats while waiting for the food-truck, during lunch or at coffee time. They were always deep and interesting, and always managed to relax me before going back to work.

I now want to thank all the people that left me happier than when I first began this three years of work. Of course, my first thought comes to my parents, and to my siblings Benoît, Quitterie and Xavier. I would be nothing without them! These thanks extend to all my family, with a special thought to my aunt Agnès and my uncle Pierre who welcomed

me several weekends.
Among the friends I met in Toulouse, I want to thank all those who made the lockdown easier: Kiki, Tassila, Marie-Noël, Hortense, Romain, Arnaud et Simon. Naturally, I also want to thank Marie-France, Jean-Antoine and their kids for easing everything, and for putting everything under control. I cannot forget my other roommates: Victor, Lara, Pierre and Gaspard. These three years would have been also harder to bear without all the people from the ENS Cachan that were on the same boat: Jérôme D., Alexandre and François in Toulouse and Antoine, Anatole and Jérôme C. outside of it. I wish you luck for the end of your PhD for those who did not defensed their PhD yet. Finally, I want to thank Fr Jean-Baptiste, Sylvain and Jean-Raphaël for their support.
Of course, I also want to thank in this paragraph all the people I met at work that I already cited in the previous paragraph, and particularly Sébastien and Jérôme. Without you, life would not be that funny.

Contents

1 Introduction **1**
- 1.1 Optimism for learning in games . . . 2
- 1.2 Bandits in metric spaces for optimization and approximation . . . 9

2 Introduction en Français **26**
- 2.1 De l'optimisme pour apprendre dans les jeux . . . 27
- 2.2 Bandits à valeur dans un espace métrique pour l'optimisation et l'approximation 35

3 Optimistic Gradient Descent Ascent in Zero-Sum and General-Sum Bilinear Games **52**
- 3.1 Introduction . . . 53
- 3.2 Zero-sum bilinear games . . . 56
- 3.3 General-sum bilinear games . . . 57
- 3.4 Illustration: Generative Adversarial Networks . . . 61
- 3.5 Conclusion . . . 64
- 3.A Comments on Theorem 3.1 . . . 66
- 3.B Proof of Theorem 3.1 . . . 67
- 3.C Proof of Theorem 3.3 . . . 79
- 3.D Proof of Theorem 3.4 . . . 83
- 3.E Proof of Theorem 3.5 . . . 84
- 3.F Proof of Proposition 3.6 . . . 86

4 Certified Multi-Fidelity Zeroth-order Optimization **89**
- 4.1 Introduction . . . 90
- 4.2 The certified MFDOO algorithm, and its cost complexity . . . 93
- 4.3 Lower bound . . . 99
- 4.4 Special case: noisy evaluations of f (a.k.a. stochastic setting) . . . 103
- 4.A Missing proofs in Section 4.3 . . . 107
- 4.B Useful lemmas . . . 109
- 4.C The special case of constant costs . . . 110
- 4.D On the characterization of $S_\beta(f,\varepsilon)$ . . . 111

5 Adaptive Approximation of Monotone Functions **114**
- 5.1 Introduction . . . 115
- 5.2 Lower bound . . . 120
- 5.3 Upper bound . . . 121

5.4 Improvement for special cases 125
5.5 Numerical experiments 131
5.6 Conclusion and future works 135
5.A Elementary properties of box-covers 137
5.B Omitted proofs 140
5.C Numerical experiments: the L^2-norm Case 154

Chapter 1

Introduction

Trust has always been the cornerstone of interactions between humans, because at the heart of every individual lies a search for meaning and truth. This fundamental principle applies universally, extending its influence even into the realm of mathematics. From the bounding of π by Archimedes over two thousand years ago up to latest precision measurements of the fundamental constants of our universe in the Planck 2018 results (Aghanim et al., 2020), mathematicians and physicist always strive for tighter and more reliable guarantees.

In our modern area, where algorithms spread through every facet of our lives, certifiable algorithms play a pivotal role in extending trust and ensuring guarantees in new technologies. Getting algorithms which converge faster is of weak interest if one cannot ascertain if it converges to a good outcome. Likewise, accurate algorithms with impressive convergence rates must need certifications to be employed in large-scale real-world software. A deeper explainability and certifiability is needed. In critical fields such as nuclear systems, technological innovation or healthcare, there is a strong inducement towards enhanced algorithmic understanding coupled with the start of more robust safety bounds.

In this thesis, we discuss these challenges through the lens of three distinct algorithms. Firstly, we provide theoretical bounds on an algorithm used in game theory, a discipline deeply in link with economics and machine learning. Then, we delve into the process of certification and improved convergence speed for an algorithm designed to tackle optimization problems with approximate evaluations, following the track of many algorithms of zeroth-order optimization, Bayesian optimization and bandits in metric spaces. Finally, we study the problem of function approximation, introducing yet another algorithm that comes with certification.

With this work, we aim to contribute to the ever-evolving quest for trust, precision and safety in the world of mathematics and beyond.

Contents

1.1 Optimism for learning in games 2

1.1.1 From zero-sum games to general-sum games 2

1.1.2 Nash equilibria . . . 3
1.1.3 Learning in games . . . 5
1.1.4 Methods for learning in games . . . 5
1.1.5 Optimism to learn in games . . . 8
1.1.6 Earlier works . . . 8
1.2 Bandits in metric spaces for optimization and approximation . 9
1.2.1 Adversarial bandits . . . 9
1.2.2 Bandits in metric spaces . . . 13
1.2.3 Methods for bandits in metric spaces . . . 14
Multi-fidelity . . . 18
A need for certification . . . 20
f-dependent bounds . . . 21
1.2.4 Sequential estimation of integrals . . . 22
1.2.5 Approximation and integral estimation of monotone functions . . 24

1.1 Optimism for learning in games

1.1.1 From zero-sum games to general-sum games

Zero-sum games are games between two players (Player 1 and Player 2), who have opposite interests. In zero-sum games, whatever the second player loses are won by the first one. Common examples include one shot games such as rock paper scissors or matching pennies. Other games use extensive form games that can be reduced to one-shot games, like Tic-Tac-Toe, or chess.

Let us first introduce finite zero-sum games. For such games, each player has a finite number of actions (or strategies). We denote by I (with cardinal n) the set of actions of Player 1, and by J (with cardinal m) the set of actions of Player 2. To each possible couple $(i,j) \in I \times J$ of actions played by both players is associated a payoff $a_{i,j}$. Player 2 needs to pay the amount $a_{i,j}$ to Player 1. In the example of the rock paper scissors game, if Player 1 plays paper (action 2) and Player 2 plays rock (action 1), then the second player needs to pay $a_{2,1} = 1$ to the first player. The higher the amount, the happier Player 1 will be, at the expense of the other player who tries to lower his loss. Note that $a_{i,j}$ can be negative, in which case Player 2 wins, after giving some negative value to Player 1. Both players need to play simultaneously, and thus cannot play according to the other's action, but should play against all possible actions of the other one.

Because the game is only determined by its payoffs $(a_{i,j})_{(i,j)\in I\times J}$, it can be represented by a matrix A. This explains why finite zero-sum games are also called matrix games. For example, the matrix associated with the rock paper scissors game is the matrix

$$A_{\mathrm{RPS}} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}.$$

Remark that if we denote by e_i the i-th vector of the canonical base, then $a_{i,j} = e_i^T A e_j$. One could then wonder if the actions of each player could be extended to any vector (x, y) in $\mathbb{R}^n \times \mathbb{R}^m$. A natural extension is to be able to play randomly between different actions. For example, assume Player 1 wants to play Rock with probability $\frac{1}{2}$ and Paper with probability $\frac{1}{2}$, in which case, its action is $\frac{\delta_{e_1}+\delta_{e_2}}{2}$ where δ_x denotes the Dirac measure at some $x \in \mathbb{R}^n$. In expectation, this action is worth $\mathbb{E}\left[\frac{\delta_{e_1}+\delta_{e_2}}{2}\right] = \frac{e_1}{2} + \frac{e_2}{2} = \left(\frac{1}{2}, \frac{1}{2}\right)$, and its expected payoff against an action y of Player 2 is $\frac{1}{2}e_1^T Ay + \frac{1}{2}e_2^T Ay = \left(\frac{1}{2}, \frac{1}{2}\right)^T Ay$. We say that such an action is a mixed strategy, on the contrary to non-random action e_i, which are called pure strategies. By abuse of notation, we define mixed strategies by their expected action, and their payoff by their expected payoff. Thus, mixed actions lay in the n-th (m-th for Player 2) simplex defined by $\Delta(n) = \{x \in \mathbb{R}^n_{\geqslant 0} : \sum_{i=1}^n x_i = 1\}$. We usually denote by $x \in \Delta(n)$ and $y \in \Delta(m)$ mixed strategies of Player 1 and Player 2 respectively. According to the previous definition, the payoff for two mixed strategies x, y is thus the scalar $x^T Ay$. Games with all strategies on the simplex are called constrained games.

Another extension is to allow the players to play any action in the whole set $\mathbb{R}^n \times \mathbb{R}^m$. Once again, the payoff after playing (x, y) is $x^T Ay$. Such games are called unconstrained. Constrained games are more popular, and present some strong properties.

For now, all the games presented are bilinear, but we can think of games with more complex payoffs. One could for example think of concave/convex payoffs (concave in the first variable and convex in the second) on compact and convex sets $\mathcal{X}$ and $\mathcal{Y}$. For now, we go back to the more classical setting of zero-sum games with mixed payoffs, due to their strong properties, to more clearly state useful definitions, and we will return later to unconstrained and concave/convex games.

A more general setting is the one of non-zero sum games, also called *general-sum games*. There, what Player 1 wins is not necessarily what Player 2 loses. Imagine for example a scenario where two friends would like to do an activity together. The first one wishes to attend a match of football, while the second is eager to go to a rugby match. Of course, both friends would prefer to go together to the same match. We could represent this setting with the following table:

$$\begin{pmatrix} (2,1) & (0,0) \\ (0,0) & (1,2) \end{pmatrix}$$

where for each couple (i, j) of actions, two coefficients $a_{i,j}$ and $b_{i,j}$ are received, one which is the payoff of the first friend, and the other the payoff of the second friend. This game is called the "Battle of the sexes". This problem can thus be modeled by two $n \times m$ matrices A and B. A general-sum game is thus denoted by its payoff matrices (A, B), or more generally by its payoffs (g_1, g_2) for non-matrix games.

1.1.2 Nash equilibria

In the case of rock paper scissors, if one player always plays "rock", the other one needs to play "paper" to win. This is the notion of best response: the best possible action

against a specific action of the other player. Best responses are not always unique. For example, in the zero-sum game $\begin{pmatrix} 1 & 3 \\ 1 & -3 \end{pmatrix}$, if Player 2 plays 1, both actions 1 and 2 are a best responses of Player 1, giving a payoff of 1 in both cases. We denote by $\mathrm{BR}_1(y)$ (resp. $\mathrm{BR}_2(x)$) the set of best responses for Player 1 (resp. Player 2) against an action y of Player 2 (respect. x of Player 1). Formally, the set of best responses is defined by $\mathrm{BR}_1(y) = \mathrm{Arg}\max_{x\in\Delta_n} x^T Ay$ and $\mathrm{BR}_2(x) = \mathrm{Arg}\max_{y\in\Delta_m} x^T By$.

By definition, a Nash equilibrium of a bilinear constrained game is a couple $(x_\star, y_\star) \in \Delta(n)\times\Delta(m)$ of strategies such that x is a best response against y and y is a best response against x. Then, for any $x \in \Delta(n)$, $y \in \Delta(m)$, $x^T Ay_\star \leqslant x_\star^T Ay_\star$ and $x_\star^T By_\star \leqslant x_\star^T By$ (this definition can be generalized to concave/convex games). In the case of zero-sum games, it means that $(x_\star, y_\star)$ is a saddle-point of the payoff function.

For one-shot games, what one really needs is not to be good against the other player, as players do not know the other's strategy beforehand, but to be good against any possible strategy. For player 1, this means to find a strategy $x_\star$ that maximizes $\max_{x\in\Delta(n)} \min_{y\in\Delta(m)} x^T Ay$. Such a strategy is called optimal for Player 1.

A game changer result in game theory is the following min-max theorem for two players zero-sum games with mixed strategies from von Neumann (1928), later proved by Nash in the case of n-player games (Nash, 1951):

Theorem 1.1 (von Neumann's min-max theorem)**.** *Let $A \in \mathbb{R}^{n\times m}$ be a payoff matrix. Then*

$$\max_{x\in\Delta(n)} \min_{y\in\Delta(m)} x^T Ay = \min_{y\in\Delta(m)} \max_{x\in\Delta(n)} x^T Ay$$

We call value of the game A, denoted by $Val(A)$ this quantity.
Moreover, both players have an optimal strategy.

Note that a generalization theorem exists for concave/convex functions on compact convex sets (Sion, 1958):

Theorem 1.2 (Sion's theorem)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be two non-empty compact convex sets, and $g : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ be a continuous and concave/convex function. Then the game possess a value:*

$$\max_{x\in\mathcal{X}} \min_{y\in\mathcal{Y}} g(x,y) = \min_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} g(x,y)$$

Moreover, both players have an optimal strategy.

We also spoke about unconstrained bilinear games, that is, games with action on the whole Euclidean space. Because the proof of Nash's and Sion's theorems are based on the compactness of the sets, one could ask oneself whether a Nash equilibrium and a value exists for such games. The answer is yes, and follows from basic linear algebra results. It can even be extended to general-sum games.

Lemma 1.3. *Let $A \in \mathbb{R}^{n\times m}$ be a payoff matrix. Then the unconstrained game has a value which is 0, and the game has at least one Nash equilibrium $(x_\star, y_\star) = (0,0) \in \mathbb{R}^n \times \mathbb{R}^m$.*

In the following, we denote by $\mathrm{Ker}(A)$ the kernel of a matrix $A \in \mathbb{R}^{n\times m}$, that is the set of all vectors $y \in \mathbb{R}^m$ such that $Ay = 0$. We also denote by $\mathrm{Im}(A)$ the image of A, that is the set of all $z \in \mathbb{R}^n$ such that there exists some $y \in \mathbb{R}^n$ with $z = Ay$.

Proof. Let $(x_\star, y_\star)$ be in $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$. Then for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$,

$$x^T A y_\star \leqslant x_\star^T A y_\star \leqslant x_\star^T A y,$$

all these terms being equal to 0. □

The previous theorem still holds for general-sum bilinear games. However, in this setting, it is not always beneficial anymore for the players to play a Nash equilibrium. For example, consider the case where $A = B = I_n$, where I_n is the identity matrix in dimension $n = m$ with ones on the diagonal and zeros everywhere else. Then, it is always beneficial for the players to both play the same vector with the highest norm possible. Both their payoffs could be arbitrarily large simultaneously. This behavior will be observed later in Chapter 3, and is similar to the one of Coarse Correlated equilibrium.

1.1.3 Learning in games

In real-life, many games are played sequentially. One, for example, is always eager to play several rounds of rock paper scissors. The winner is then the competitor who won the most rounds after a previously set number of rounds. In this scope, one would like to see how the strategy of one player evolves while playing the game. *Learning in games* is the study of how players learn to play a game, to what action or set of actions the strategy of each player converges, or at what speed this convergence occurs. From now on, we consider infinite stage games.

Let $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a payoff function. The protocol is the following: Simultaneously, Player 1 chooses an action $x_1 \in \mathcal{X}$ and Player 2 chooses an action $y_1 \in \mathcal{Y}$. Then, Player 2 pays $g(x_1, y_1)$ to Player 1, and both players observe each other's action (so far, we are still playing a one-shot game, things will change now). Now, at time t, both players have access to the total current history of gradients $h_{t-1} = (A^T x_1, A y_1, \ldots, A^T x_{t-1}, A y_{t-1})$, and choose their next action $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$ accordingly. They then receive the payoff $g(x_t, y_t)$. We denote by $\sigma = (\sigma_t)_{t\geqslant 1}$ a strategy for Player 1, and by τ a strategy for Player 2. For all $t \geqslant 1$, σ_t is a function that associate to any possible value $h_{t-1} \in (\mathbb{R}^n \times \mathbb{R}^m)^{t-1}$ of the history an action $x_t \in \Delta(n)$.

1.1.4 Methods for learning in games

In this section, we study multi-stage games, and the models that the players can use to converge to a Nash equilibrium of the game.
The first method we could think of is for each player to play a best response against the empirical average action of its opponent. That is, at each time t, the strategy σ_t of Player 1 is to play any x_{t+1} in $\mathrm{BR}_1\left(\frac{1}{t-1}\sum_{s=1}^{t-1} y_s\right)$. This algorithm introduced by Brown

(1951) is called Fictitious Play. Note that the best response may not be defined in unconstrained problems, but that Fictitious Play has hardly been studied in this setting. It is known to converge *in average* to a Nash equilibrium for two-player matrix zero-sum games (Robinson, 1951). However, the speed of convergence is polynomial with t, and may be really slow as the number of actions increases (Daskalakis and Pan, 2014), on the contrary to what Karlin expected (Karlin, 2003, Section 6.6). Worse, Fictitious play *does not show last-iterate convergence* at least for games without a pure Nash equilibrium, as the actions of each player alternate between different pure strategies. There exists a stochastic version of Fictitious Play introduced in (Fudenberg and Kreps, 1993) that is known to possess *last-iterate convergence* since the work of Hofbauer and Sandholm (2002).

Various other algorithms have been proposed to find saddle points. Examples include those by Arrow et al. (1958); Rockafellar (1976); Nemirovski and Yudin (1983); Freund and Schapire (1999). We focus here on the variant of Gradient Descent Ascent called Optimistic Gradient Descent Ascent, or OGDA for short, with constant step size $\eta > 0$.

Algorithm 1 Gradient Descent Ascent

Input: A function g
Initialization: A step size $\eta > 0$, an initialization $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$
for $t = 0, \dots,$ **simultaneously do**
$x_{t+1} = \Pi_{\mathcal{X}} \left(x_t + \eta \frac{\partial g}{\partial x}(x_t, y_t) \right)$
$y_{t+1} = \Pi_{\mathcal{Y}} \left(y_t - \eta \frac{\partial g}{\partial y}(x_t, y_t) \right)$
end for

First recall the basic Gradient Descent Ascent algorithm (also denoted GDA, Algorithm 1), where at each iteration t both players run a gradient step. Generative Adversarial Networks (Goodfellow et al., 2014; Biau et al., 2020), also called GANs, are two deep neural networks that play a game against each other with the goal for the first one to approximate a probability distribution, and for the second to differentiate true data from generated data. Most GANs currently use (stochastic) GDA under the hood. In several zero-sum games, GDA was acknowledged to converge *in average* to the solution of the min-max problem (Nedić and Ozdaglar, 2009). However, it is well known that even for the simple problem $g(x, y) = xy$ with $\mathcal{X} = \mathcal{Y} = \mathbb{R}$, GDA may exhibit a cyclic behavior and its *last iterate may diverge*.

Proposition 1.4. *Let $A \in \mathbb{R}^{n \times p}$ be a matrix payoff of an unconstrained bilinear zero-sum game. Then, the GDA dynamic may not converge to a Nash equilibrium of the game with matrix payoff A.*

Proof. For the "Matching Pennies" game, where $n = m = 1$ and $A = (1)$, $x^T A y$ is simply xy. We then obtain:

$$x_{t+1} = x_t + \eta y_t \text{ and } y_{t+1} = y_t - \eta x_t$$

which implies $x_{t+1}^2 + y_{t+1}^2 = (x_t^2 + y_t^2)(1 + \eta^2)$. It means that $\|(x_t, y_t)\| \xrightarrow[t \to \infty]{} +\infty$ as soon as $(x_0, y_0) \neq (0, 0)$.

□

Proposition 1.4 as well as a last-iterate convergence results can be extended to Mirror Descent Ascent (Shalev-Shwartz and Singer, 2006), an adaption of the mirror descent algorithm from Nemirovski and Yudin (1983) to the min-max setting. The mirror descent algorithm updates the strategy with a gradient step in a "mirror space" that depends on some fixed regularizer (a strongly-convex function). When the regularizer is the Euclidean norm, Mirror Descent Ascent reduces to the GDA, and when it is the negative entropy on the simplex, we recover the famous Multiplicative Weight Update (MWU).
A nice variant of Mirror Descent Ascent to overcome this cyclic behavior is the Optimistic Mirror Descent Ascent algorithm. It reduces to the Optimistic Gradient Descent Ascent algorithm (OGDA, Algorithm 2) when used with the Euclidean norm.

Algorithm 2 Optimistic Gradient Descent Ascent

Input: A function g
Initialization: A step size $\eta > 0$, an initialization $(x_0, y_0, x_{-1}, y_{-1}) \in \mathbb{R}^{(n+m+n+m)}$
for $t = 0, \ldots,$ **simultaneously do**
$x_{t+1} = \Pi_{\mathcal{X}} \left(x_t + 2\eta \frac{\partial g}{\partial x}(x_t, y_t) - \eta \frac{\partial g}{\partial x}(x_{t-1}, y_{t-1}) \right)$
$y_{t+1} = \Pi_{\mathcal{Y}} \left(y_t - 2\eta \frac{\partial g}{\partial y}(x_t, y_t) + \eta \frac{\partial g}{\partial y}(x_{t-1}, y_{t-1}) \right)$
end for

An example of the cyclic behavior of the Gradient Descent Ascent algorithm and of the convergence of its optimistic counterpart can be seen in Figure 1.1 on the Matching Pennies game with payoff matrix $A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. For mixed strategies, the Nash equilibrium of the game is to play the uniform distribution over the set of possible actions for both players.

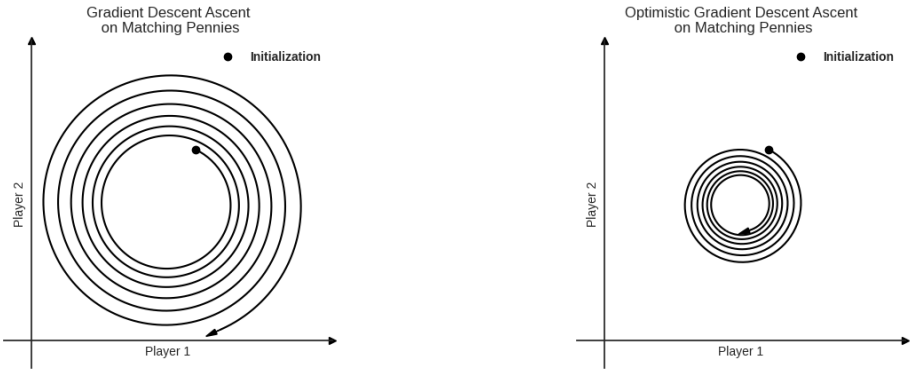


Figure 1.1: Comparison of the mirror descent ascent and its optimistic counterpart for Matching Pennies, for $\eta = 0.02$.

1.1.5 Optimism to learn in games

In the following, we consider the unconstrained bilinear case, where $\mathcal{X}$ and $\mathcal{Y}$ are Euclidean spaces, *i.e.*, there exist integers $n, m \geqslant 1$, such that $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$. For this reason, the regularizer we take is the Euclidean norm, and we use OGDA. We stick here to the standard case where the weights of the gradients of the stage $t-1$ are half the weights of the gradients of the stage t, but other choices are possible. One can look at the work of Peng et al. (2020) for the behavior for other coefficients.

This *optimistic* algorithm has been introduced by Korpelevich (1976); Popov (1980), and rediscovered in the GAN literature decades later (Daskalakis et al., 2018). A challenge is then to determine when does OGDA converge to an equilibrium of the game. In the setting of learning in games, an algorithm is said to be optimistic, when it uses an intermediary state to compute the gradient: the point at time $t+1$ is computed with the use of the gradient at time t (like regular algorithm so far) *and* the gradient at some time $t+1/2$.[1] The optimistic gradient descent ascent algorithm is an example of such a algorithm. Moreover, it is a single-call algorithm: we do not compute the gradient at another time $t+1/2$, but we use the previously computed gradient to approximate it. Here, we do so with the use of the gradient at time $t-1$.

In Chapter 3, we study the convergence of OGDA for unconstrained bilinear games. We provide the exact best ratio of convergence as a function of the step size η, and find the step size with the fastest convergence. A study of OGDA for general-sum games is then done. It shows that OGDA converges to a suitable outcome for a large set of games. This result is then used to speed up the convergence in a zero-sum problem by introducing OGDA on a well picked general-sum game. This interestingly shows that general-sum games can be used to improve algorithms designed for min-max games. We then look at the experimental results, whether in the bilinear setting, or in the concave/convex setting.

1.1.6 Earlier works

The optimistic version of the Gradient Descent algorithm was first seen in a paper by Popov (1980). Rakhlin and Sridharan (2013) were the first to introduce the optimistic mirror descent algorithm (OMD), and showed that it is a no-regret algorithm. Optimistic Gradient Descent Ascent is a particular case of OMD, for which the regularizer used is the Euclidean norm.

In the context of GANs, a first study of OGDA for bilinear games was done by Daskalakis et al. (2018). They proved last-iterate convergence of OGDA, in the sense that for each $\varepsilon > 0$ one can find $\eta > 0$ and T large enough such that (x_T, y_T) (as defined in Algorithm 2) is ε-close to a Nash equilibrium. Later, Liang and Stokes (2019), proved that in the case of a square invertible matrix A, OGDA converges to $(0,0)$ at exponential speed for some specific value of the step size η. Gidel et al. (2019) studied GANs through variational inequalities, with a convergence ratio not better than the one proved in (Liang

[1] One should be careful: this notion of optimism is different than the one used in the bandits setting. This second sort of optimism will be discussed in Section 1.2

and Stokes, 2019).

Another variant of GDA is the Extra-Gradient method (EG) introduced by Korpelevich (1976). Mokhtari et al. (2020a) showed that OGDA and EG can be seen as approximations of the Proximal Point method, and proved the exponential convergence of the three algorithms for square invertible bilinear games and a particular value of η.
Zhang and Yu (2020) studied deeply GDA, EG, ODGA, and a heavy ball method. They gave in particular the optimal step size $\eta^\star$ and its associated geometric ratio $\lambda^\star$ for OGDA w.r.t the payoff matrix A. Another related work is the one of Hsieh et al. (2019) who studied among other things the stochastic setting for EG and OGDA. The authors also unearth the tight link between the OGDA and EG algorithms.

Beyond bilinear games, several papers considered OGDA for concave/convex games. In particular, Daskalakis and Panageas (2018) studied the stability of fixed points of the dynamics, and Abernethy et al. (2018) improved the rate of convergence for some specific concave/convex games. A later paper by Lin et al. (2020) reached for another algorithm of their creation the lower-bound for strongly-concave/strongly-convex games of Ibrahim et al. (2020) (later generalized in Hadiji et al. (2023)), up to some logarithmic factors. However, the non-concave/non-convex setting exhibits problematic properties: Hsieh et al. (2021) showed there may exist attractors of OGDA containing no stationary points, and Daskalakis et al. (2021) showed that finding an approximate local min-max equilibrium is PPAD-hard. Nevertheless, in a recent paper, Diakonikolas et al. (2021) showed positive results in a new class of non concave/non convex min-max problems.

Many studies were also done in compact settings, that is when $\mathcal{X}$ and $\mathcal{Y}$ are compact convex sets. Daskalakis and Panageas (2019) proved the convergence of an Optimistic version of Multiplicative Weight Updates on the simplex, for an appropriate value of the learning rate. Later, Wei et al. (2021) improved their results by showing the convergence for any learning rate smaller than or equal to some constant. Finally, in a more recent paper, Anagnostides et al. (2022) considered the convergence of a projected version of OGDA for general-sum games, where after each iteration, a projection on the simplex is applied.

1.2 Bandits in metric spaces for optimization and approximation

1.2.1 Adversarial bandits

In the previous section, we looked at games where both players are concerned about their gains. We now assume that the second player is oblivious of what is happening. Player 2 will now be *nature* or *an environment*: it does not try to maximize its gain, but plays round after round without a (perceptible) strategy. Let us take the example of a man who needs to choose every morning whether to take his umbrella or not before he goes out, without knowing the weather for the day to come. Every day, the environment can *be in* (instead of *choose between*) two different states: sunny or rainy. Of course, the goal

of this man is to take his umbrella when it rains and leave it when its sunny. He will act according to actions that maximize his payoff. However, the environment does not care about the outcome of the game.

The model that we use is the following: at each time t, the player (often called the learner in this setting) picks some action $x_t \in \mathcal{X}$ which follows from some previously chosen strategy σ_t, and he then receives a reward r_t. For all t, σ_t takes the history h_{t-1} of queries $x_1, \dots, x_{t-1}$ and of rewards $r_1, \dots, r_{t-1}$ into account in order to choose $x_t = \sigma_t(h_{t-1})$. $\sigma = (\sigma_t)_{t \in \mathbb{N}^*}$ is called a *behavior strategy*, or simply *strategy*. Note that the strategy σ can be deterministic or stochastic.

Different settings were studied in adversarial bandits. The most well-known of all is the *multi-armed bandits* (Thompson, 1933; Robbins, 1952). In multi-armed bandits, $\mathcal{X}$ is a finite set with n elements, x_t is an action (or arm) in $\mathcal{X}$ chosen by the learner, and the reward r_t follows a law μ_{x_t}. The aim of the learner is to find the $x \in \mathcal{X}$ such that μ_x has the highest expectation. Assumptions must be verified by the μ_x for all actions $x \in \mathcal{X}$. Often times, the μ_x are considered as independent Bernouilli variable with some parameter $p_x \in (0,1)$.[2] More general cases examine any independent bounded or even subGaussian variables.[3] In the more general case of bandits convex optimization (Flaxman et al., 2005), the reward r_t is of the form $g_t(x_t)$ for some concave functions $(g_t)_{1 \leqslant t}$ unknown to the learner, with uniformly bounded value.

Note that at the end of each time t, the learner only receives the feedback $g_t(x_t)$. In the easier problem of learning with experts, the learner can have full knowledge of the reward function g_t. In this full-feedback setting, the most famous methods and algorithms are all first-order methods: they make use of the gradient of the function. Examples include the Online Gradient Descent of Zinkevich (2003), and its generalization the Online Mirror Descent (Beck and Teboulle, 2003; Shalev-Shwartz and Singer, 2006), as well as the Exponential Weight Algorithm, another name for the Multiplicative Weight Update (Littlestone and Warmuth, 1994; Vovk, 1990).[4] Note that most of these methods have a strong resemblance with the one presented in Section 1.1.4. This comes from the fact that strategies for both settings are inspired by first-order methods from convex optimization. On the contrary to online learning, in bandits problem, one can only use *zeroth-order* methods: the learner does not have access to the gradient of the function, only to the (sometimes stochastic) evaluation of the function f.

In the bandits convex optimization problem, any learner wishes to find the optimal actions $x_1^\star, \dots, x_T^\star$ that maximize their total expected payoff $\mathbb{E}\left[\sum_{t=1}^T g_t(x_t)\right]$ against all possible state of the environment, that is, against all adversarial sequence $(g_t)_{t \in \mathbb{N}^*}$ of uniformly bounded concave functions that he could possibly face. Assume for simplicity that g_t has value in $[0,1]$ and that its maximal value is 1. We denote by $x_t^\star$ any maximal argu-

[2] A variable X is a Bernouilli variable if $\mathbb{P}(X=1) = 1 - \mathbb{P}(X=0) = p$.

[3] A real-valued random variable X is v-subGaussian if $\mathbb{E}[\exp(\lambda X)] \leqslant \exp(\lambda^2 v/2)$ for all $\lambda \in \mathbb{R}$.

[4] This algorithm possesses many other aliases, such as the Hedge algorithm, the Weighted Majority algorithm, and many more.

ment of g_t (with value equal to 1). Then, a learner would like to compare himself to this optimal *a posteriori* payoff, which is worth T in our example. Thus, it would deplore an expected loss of earnings of $\sum_{t=1}^{T}\mathbb{E}\left[(g_t(x_t^\star)-g_t(x_t))\right]=T-\sum_{t=1}^{T}\mathbb{E}\left[g_t(x_t)\right]$ for the sequence of reward functions $(g_t)_{1\leqslant t\leqslant T}$. Now, imagine that g_t takes independently of the previous history and uniformly at random the value $g_t : x\to x$ or $g_t : x\to 1-x$. Then the learner has no way to guess with any method the best strategy σ, and will receive a payoff of $1/2$ in expectation at each round. This gives a loss of earnings of $T/2$. It is of the same order of T, the worst loss of earnings possible. This explains why in the bandits literature all the results used a bound on the regret: the loss of earnings compared to the *best fixed action*.

We denote by R_T^c the *cumulative regret* defined by

$$R_T^{\mathrm{c}}=\mathbb{E}\left[\max_{x\in\mathcal{X}}\sum_{t=1}^{T}g_t(x)-\sum_{t=1}^{T}g_t(x_t)\right].$$

We also define the *simple regret*: the regret of the $(T+1)^{\text{th}}$ guess after having observed T rewards of the game, where the T query points were chosen by the learner.

$$R_T^{\mathrm{s}}=\mathbb{E}\left[\max_{x\in\mathcal{X}}g_{T+1}(x)-g_{T+1}(x_{T+1})\right].$$

For a constant payoff g (that is $g_t=g$ for all $t\leqslant T+1$), the cumulative regret equals the sum of all the regrets for time $1\leqslant t\leqslant T$. Hence, in order to minimize this objective, a player should always be careful about the action he chooses at any time t and be careful it does not have a strong regret compared to the best fixed action. On the contrary, for the simple regret, only the last action matters. Thus, the learner can freely explore regions with really bad rewards to get more information, without taking any penalties. For this reason, the cumulative regret is a stronger notion than the simple regret up to a rescaling factor of $1/T$. When the payoffs $(g_t)_{t\in\mathbb{N}}$ are independent and identically distributed, this is formally expressed by the following inequality: $\mathbb{E}[R_T^{\mathrm{s}}]\leqslant\frac{\mathbb{E}[R_T^{\mathrm{c}}]}{T}$.

Now, let us look at an exploration-exploitation strategy with good performances on the cumulative regret. The study of this algorithm will help us better understand the motives behind the algorithms we will cross later in more difficult bandits settings.

The upper confidence bound algorithm: We are now going to present and intuitively explain the Upper Confidence Bound algorithm (Auer et al., 2002) for the multi-armed bandits setting. For simplicity, we consider a multi-armed bandits problem where each action $x\in\mathcal{X}$ is linked to an unknown law μ_x with support in $[0,1]$ and expectation $f(x)$. With this setup, f is a function from the finite set $\mathcal{X}$ to the bounded set $[0,1]$, but it will later be extended to infinite input sets. In order to solve multi-armed bandits, a first method would be a "Fictitious Play"-like algorithm that chooses at each time t the action x with the best empirical average $\widehat{f}^t(x)$. This means that at all time t, $x_t=\arg\max_{x\in\mathcal{X}}\widehat{f}^t(x)$ with $\widehat{f}^t(x)=\frac{\sum_{s=1}^{t}\mathbb{1}_{\{x_s=x\}}g_s(x_s)}{T_{x,t}}$ where $T_{x,t}=\sum_{s=1}^{t}\mathbb{1}_{\{x_s=x\}}$ and $g_s(x_s)\sim\mu_{x_s}$. This dull algorithm, called Follow-The-Leader, fully exploits the data, but does not explore. Hence, when wrong, it does not ask for more information, but stays with its wrong perception of

the environment and keeps receiving sub-optimal reward, and thus, keeps increasing its regret.

In order to add exploration, an idea would be to take $x_t = \arg\max_{x\in\mathcal{X}} u_{x,t}$ where $u_{x,t}$ is of the form $u_{x,t} = \widehat{f}^t(x) + R(T_{x,t})$ for some decreasing function R. R plays the role of a regularizer. The question is now to find the good R. Indeed, if it decreases too slowly, we will still be in the previous case with no exploration, but if it decreases too quickly, the algorithm will never exploit and choose actions that are close to be uniformly distributed.

The choice of this extra-term is solved thanks to Hoeffding's inequality, a concentration inequality for bounded variable.[5] Indeed, it tells us that for any s and any action $x \in \mathcal{X}$,

$$\mathbb{P}\left(f(x) - \widehat{f}^s_{\text{emp}}(x) \geqslant \varepsilon\right) \leqslant \exp(-2s\varepsilon^2)\,,$$

where $\widehat{f}^s_{\text{emp}}(x)$ is the empirical mean after choosing action x s times. Thus,

$$\mathbb{P}\left(f(x) - \widehat{f}^t(x) \geqslant \varepsilon\right) \leqslant \exp(-2T_{x,t}\varepsilon^2)\,,$$

Thus for $\delta = t^{-4}$,

$$\mathbb{P}\left(f(x) - \widehat{f}^t(x) \leqslant \sqrt{\frac{\ln(1/\delta)}{2T_{x,t}}}\right) \geqslant 1 - \mathbb{P}\left(\exists s \in \{1,\dots,t\},\ f(x) - \widehat{f}^s(x) \geqslant \sqrt{\frac{\ln(1/\delta)}{2s}}\right) \geqslant 1 - t^{-3}\,.$$

We can have the same type of inequality for $\widehat{f}^t(x) - f(x)$ instead of $f(x) - \widehat{f}^t(x)$. It allows us to upper bound with high probability the mean of each of the actions according to the empirical mean, where the bound depends increasingly on the number t of round. This gives us the Upper Confidence Bound Algorithm (UCB for short, Algorithm 3).

Algorithm 3 Upper Confidence Bound

Input: A set of actions $\mathcal{X}$.
Initialization: Set $T_{x,0} = 0$ for all x and $u_{x,0} = +\infty$.
for $t = 0,\dots,$ **do**
  Pick $x_t = \arg\max_{x\in\mathcal{X}} u_{x,t-1}$;
  Observe r_t;
  Set $T_{x_t,t} = T_{x_t,t-1} + 1$, $\widehat{f}^t(x_t) = \frac{T_{x,t}\widehat{f}^{t-1}(x_t) + r_t}{T_{x,t}+1}$ and $u_{x_t,t} = \widehat{f}^t(x_t) + \sqrt{\frac{2\ln(t)}{T_{x,t}}}$
  For all other action x, set $T_{x,t} = T_{x,t-1}$, $\widehat{f}^t(x) = \widehat{f}^{t-1}(x)$ and $u_{x,t} = u_{x,t-1}$.
end for

[5] Hoeffding's inequality for independent variables $X_1,\dots,X_T$ with values between a and b states that

$$\mathbb{P}\left(\sum_{t=1}^{T} X_t - T\mathbb{E}[X_1] \geqslant \varepsilon\right) \leqslant \exp\left(\frac{2\varepsilon^2}{T(b-a)^2}\right)\,.$$

This algorithm is an example of *optimism* in bandits, because it maximizes an upper bound $u_{x,t}$ of the mean $\widehat{f}^t(x)$ for every action x. Optimism[6] is often used in diverse branch of bandits problem as we shall see later.

Now, let us present the regret bounds of this algorithm. They are of two types. The first type of bounds are *worst-case bounds* with regard to the number n of actions and to the number T of rounds. It states that the cumulative regret of UCB verifies

$$R_T^c = O\left(\sqrt{nT\log(T)}\right).$$

This is true whatever the distribution of the rewards for each action.
The second type of bounds are *distribution-dependent bounds*. Let $\Delta_x = \max_{x' \in \mathcal{X}} f(x') - f(x)$ be the gap between the maximum mean reward and the mean reward of action x. Then the regret of UCB satisfies

$$R_T^c \leqslant 2n + \sum_{x:\Delta_x > 0} \frac{8\log(T)}{\Delta_x}.$$

This bound can be far smaller than the previous one for some specific distributions μ_x, as soon as Δ_x is large for all x. This is the main advantage of the distribution dependent bounds. Most of the time, distribution dependent bounds are harder to show, but they always give a better insight on the behavior of any algorithm by giving specific bounds to all different distributions to the contrary of worst-case bounds that treat all different distributions as one.

The difference between worst-case and distribution free bounds exists also in other settings. This is, for example, the case for prediction with expert advice. For example, De Rooij et al. (2014) developed an algorithm called FlipFlop, which achieves optimal worst-case bound of the same order than parameter free algorithms like AdaHedge (Erven et al., 2011) or CBMS Cesa-Bianchi et al. (2007), while providing better guarantees whenever the data is easy (almost as good as Follow The Leader): it adapts to the data in question.

We will now introduce bandits in metric spaces, and see how these distribution-free bounds apply in this setting.

1.2.2 Bandits in metric spaces

A metric space is a set $\mathcal{X}$ equipped with a distance d. A distance function is a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which verifies the following properties:

- $d(x,x) = 0$ for all $x \in \mathcal{X}$ (the distance from a point to itself is 0);
- for all $x, y, z \in \mathcal{X}$, $d(x,z) \leqslant d(x,y) + d(y,z)$ (it respects the triangular inequality);

[6] Optimism in bandits does not have the same background and meaning as optimism in game theory (as, for example, the optimism in the Optimistic Gradient Descent Ascent algorithm), and should not be mixed up.

- $d(x,y) \geqslant 0$ for all $x,y \in \mathcal{X}$ (it is positive);
- $d(x,y) = d(y,x)$ for all $x,y \in \mathcal{X}$ (it is symmetric).

The most usual metric spaces are Euclidean spaces with the distance induced by the Euclidean norm.

Let $(\mathcal{X},d)$ be a metric space with $\mathcal{X}$ compact. The setting of bandits in metric spaces (Kleinberg et al., 2008, 2019) is the following. Let $f : \mathcal{X} \to \mathbb{R}$ be a function. At each time step t, the learner chooses an action x_t, and receives r_t which follows a distribution μ_{x_t} with mean $f(x_t)$. In the following, we consider the simple regret $R_T^{\mathrm{s}} = \mathbb{E}\left[\max_{x\in\mathcal{X}} f(x) - r_{T+1}\right]$.

When the reward r_t is deterministic and equals $f(x_t)$, minimizing the simple regret R_T^{s} after some set number of round T turns out to evaluate the function f at T different points before choosing a point x_{T+1} at which the learner thinks that f is maximized. Indeed, in this setup, the simple regret is $R_T^{\mathrm{s}} = \max_{x\in\mathcal{X}} f(x) - f(x_{T+1})$, and to minimize it means to maximize $f(x_{T+1})$. Then the bandits in metric spaces join the framework of zeroth-order (or black-box) optimization, and of Bayesian optimization.

In the finite case (that is to say when $\mathcal{X}$ is a finite set), there is no need for strong assumptions on the function f. Nevertheless, it is not anymore the case as soon as the number of actions is infinite. Otherwise, to find the maximum of any function is impossible and the simple regret can be as big as possible. What we would like is to know that $f(x')$ is close to $f(x)$ for any x' in the neighborhood of x. For this reason, we assume next that there exists a constant $L > 0$ such that for any two $x,x' \in \mathcal{X}$, $|f(x) - f(x')| \leqslant L \cdot d(x,x')$. This condition is called Lipschitz continuity of a function. A function f that is Lipschitz continuous for some L is said to be L-Lipschitz.

This setting can be generalized in several ways.
First, the reward r_t may be stochastic. Several hypotheses exist, but the most used one is that the distribution μ_x of the reward has bounded support (with mean $f(x)$) or is uniformly subGaussian for all $x \in \mathcal{X}$.
Another generalization is to replace the maximization in the simple regret by any other functional $J : (\mathcal{X} \to \mathbb{R}) \to \mathbb{R}$ (see for example the introduction of Kiefer (1957)). Then the goal of the learner would be to approximate $J(f)$ for some known J by first evaluating f at T different values, and then outputting its best guess J_T of $J(f)$. This setting is the one of numerical analysis (Stoer et al., 1980). The learner then gets a simple regret of $|J(f) - J_T|$. Besides the maximum of f, examples of such functionals include the median of the function or its integral. We will come back to this last problem later, but we will first focus on the usual simple regret with maximization.

1.2.3 Methods for bandits in metric spaces

Currently, we assume that the learner wants to maximize some unknown L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$ defined on a compact set $\mathcal{X}$, while being able to evaluate f at any point.

This setting can also be called the *single-fidelity* settings, in comparison to the multi-fidelity setting where the learner has access to α-fidelity functions of f for diverse α (an approximation of f with a known maximal bias α). The multi-fidelity setting will be explained later in greater detail. On the contrary, in the single-fidelity setting, the unknown function f can be evaluated perfectly. Here, $\mathcal{X}$ is a compact subset of $\mathbb{R}^n$ for some finite $n > 0$. The learner also has knowledge of the real number L (which is an upper bound of, and not necessarily equal to, the smallest real number ℓ such that f is ℓ-Lipschitz).

This single fidelity setting has a rich literature within deterministic or Bayesian frameworks. Next we provide a (non-comprehensive) subset of references to these related works. Optimization algorithms *without certificates* have a very long history, in convex optimization (Nesterov, 2004; Boyd and Vandenberghe, 2004; Bubeck, 2015), non-convex optimization (Hansen et al., 1992a,b; Jain and Kar, 2017), Bayesian optimization (Garnett, 2023), stochastic optimization (Spall, 2003; Bonnans, 2019), or bandits optimization (Munos et al., 2014; Slivkins et al., 2019). Among the algorithmic techniques that are closest to Chapter 4, we can mention the Piyavskii-Shubert algorithm (Piyavskii, 1972; Shubert, 1972) for zeroth-order Lipschitz optimization, as well as discretized variants such as the branch-and-bound algorithm of Perevozchikov (1990) or the DIRECT algorithm by Jones et al. (1993), to name a few. More recently several variants were also derived in the bandits community. Examples of such bandits algorithms for perfect or noisy (stochastic) evaluations of f include DOO (Munos, 2011), HOO (Bubeck et al., 2011a), (Sto)SOO (Munos, 2011; Valko et al., 2013), POO (Grill et al., 2015), which are all based on a hierarchical partition of the input domain, as is the c.MF-DOO algorithm of Chapter 4. Another discretization approach (yet computationally more challenging) is the Zooming algorithm for general metric spaces (Kleinberg et al., 2008, 2019). We refer the reader to Munos et al. (2014); Slivkins et al. (2019) for further details and references on bandits algorithms.

Also close to our work is the Bayesian optimization literature, because of its rich contributions to the multi-fidelity setting. For single-fidelity optimization, we can mention the seminal work of Kushner (1964), the use of the expected improvement function introduced in Močkus (1975), and the EGO algorithm of Jones et al. (1998), together with convergence rates in Bull (2011). The kriging community also addressed global optimization with noisy observations (Forrester et al., 2006; Picheny et al., 2013). In Srinivas et al. (2010, 2012) the authors design and study a Gaussian Process-based bandits algorithm (GP-UCB) and derive regret bounds both under Bayesian and deterministic assumptions on the underlying function f. This algorithm was later adapted in Contal et al. (2013) for the case of sequential mini-batch queries in parallel computing. A detailed review of the Bayesian optimization literature can be found in Garnett (2023).

To introduce the context, we want to present in greater details two algorithms of the previously listed ones. The first one is called Hierarchical Optimistic Optimization (HOO), and was introduced in Bubeck et al. (2011a). It is a branch-and-bound algorithm (a classic technique in zeroth-order optimization) based on the Upper Confidence Algorithm described in Section 1.2.1.
Before giving the idea of how HOO works, we first need to introduce the tree-based structure used in such branch-and-bound algorithms. A tree based structure $(X_{h,i})_{h\in\mathbb{N}^*, i\in\{0,\dots,K^h-1\}}$

is based on a K-ary tree $\mathcal{T} = (h,i)_{h\in\mathbb{N},i\in\{0,\dots,K^h-1\}}$ (a tree where all nodes have K children) for which each node is linked to a region of the whole space $\mathcal{X}$. Each node is associated with a region $X_{h,i}$ which verifies that $X_{0,0} = \mathcal{X}$ and for all (h,i), the regions $(X_{h+1,j})_{(h+1,j)\in\mathcal{C}(h,i)}$ forms a partition of $X_{h,i}$, where $\mathcal{C}(h,i)$ is the set of children of (h,i) in $\mathcal{T}$. A region $X_{h,i}$ of the tree is also called a *cell*. To each cell $X_{h,i}$ is associated a *representative* $x_{h,i}$ of the cell: an element of the cell that represents it. This representative can be an arbitrary point in the cell, but can more cleverly be chosen as the barycenter of the cell.

Now, let us present the idea behind HOO: at each time step, HOO goes through the tree from the root to one of the leaves, always choosing the child with the highest upper-confidence bound, with a formula strongly inspired by the UCB algorithm. It then queries a point in the according region of the space $\mathcal{X}$ and updates the upper bound of all ancestors of this leaf. In order to present an upper bound on the regret of HOO, we first need to introduce some definitions.

Definition 1.5 (Optimal points of a function)**.** For any $\varepsilon \geqslant 0$, we denote the set of ε-optimal points of f by $\mathcal{X}_\varepsilon := \{x \in \mathcal{X} : f(x_\star) - f(x) \leqslant \varepsilon\}$, where $f(x_\star)$ is any maximum of the function f.

Definition 1.6 (Packing number)**.** For any $\varepsilon > 0$, the ε-packing number $\mathcal{N}(\mathcal{X}',\varepsilon)$ of a subset $\mathcal{X}'$ of $\mathcal{X}$ is the largest number k of ε-separated points $x'_1,\dots,x'_k \in \mathcal{X}'$, that is, such that $\left\|x'_i - x'_j\right\| > \varepsilon$ for all $i \neq j \leqslant k$.[7] See Figure 1.2 for a visual example.

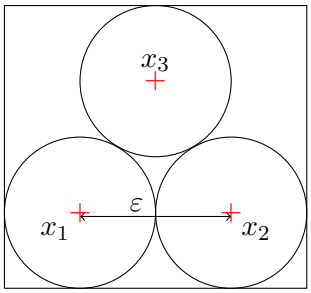


Figure 1.2: Example of an ε-packing number

Definition 1.7 (Near optimality dimension)**.** The near optimality dimension of f is the smallest integer n_f such that there exists $C_f > 0$ which verifies for any $\varepsilon > 0$ that the packing number $\mathcal{N}(\mathcal{X}_\varepsilon,\varepsilon)$ is smaller than $C_f\varepsilon^{n_f}$.

To summarize, the r-packing number of a set $\mathcal{X}'$ is the maximal number of elements at most distant of r that one can put in the set $\mathcal{X}'$, and the near optimality dimension gives an order of the rate at which the packing number of the region $\mathcal{X}_\varepsilon$ decreases with ε. The near optimality dimension of any function in $\mathcal{X}$ is smaller than or equal to n.

With these new definitions, we can present the worst-case bound for HOO given in Bubeck et al. (2011a).

[7] By convention, $\mathcal{N}(\mathcal{X}',\varepsilon) = 0$ if $\mathcal{X}'$ is empty. Note also that $\mathcal{N}(\mathcal{X}',\varepsilon) < +\infty$ since $\mathcal{X}$ is compact.

Theorem 1.8. *Let n_f be the near optimality dimension of f. Then for any $n' > n_f$, the cumulative regret of HOO after T steps is bounded as*

$$\mathbb{E}[R_T] \leqslant \gamma T^{(n'+1)/(n'+2)} (\ln T)^{1/(n'+2)}$$

for some constant γ.

Let us now focus on another algorithm, Deterministic Optimistic Optimization (DOO) introduced by Munos (2011). DOO is also a branch-and-bound algorithm based on optimism. Once again, DOO depends on a tree-based structure $(X_{h,i})_{h\in\mathbb{N}^*, i\in\{0,\dots,K^h-1\}}$ described for HOO. Let us assume that the diameter of each region $\mathcal{X}_{h,i}$ is bounded by $R\delta^h$ for some real number R and some $\delta \in (0,1)$. With this in mind, we can now present DOO in Algorithm 4.

Algorithm 4 Deterministic Optimistic Optimization

Inputs: K, $(X_{h,i})_{h\in\mathbb{N}, i\in\{0,\dots,K^h-1\}}$, $(x_{h,i})_{h\in\mathbb{N}, i\in\{1,\dots,K^h-1\}}$, δ, R, L
Initialization: Let $t \leftarrow 1$ and $\mathcal{L}_1 \leftarrow \{(0,0)\}$
Pick the first node $(h^*, i^*) \leftarrow (0,0)$
for *iteration* $= 1, 2, \dots, T$ **do**
  for all child (h^*+1, j) of (h^*, i^*) **do**
    Let $t \leftarrow t+1$ and $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{(h^*+1, j)\}$
    Pick the query point $x_t \leftarrow x_{h^*+1,j}$ and observe $f(x_t)$
  end for
  Remove (h^*, i^*) from $\mathcal{L}_t$
  Let $(h^*, i^*) \in \arg\max_{(h,i)\in\mathcal{L}_t} f(x_{h,i}) + LR\delta^h$
end for
Output the recommendation $x_T^* \in \arg\max_{1\leqslant t\leqslant T} f(x_t)$

At each time t, DOO maintains a queue $\mathcal{L}_t$ that contains all the current leaves of the tree to explore. It picks among the elements (h,i) of $\mathcal{L}_t$ the one that maximizes $f(x_{h,i}) + LR\delta^h$. We denote it by (h^*, i^*). Note that for any (h,i) and any $x \in \mathcal{X}_{h,i}$, $f(x)$ is smaller than or equal to $f(x_{h,i}) + LR\delta^h$, because f is L-Lipschitz and the diameter of $\mathcal{X}_{h,i}$ is bounded from above by $R\delta^h$. Thus, the quantity $f(x_{h,i}) + LR\delta^h$ is an upper bound of f on the region $\mathcal{X}_{h,i}$. To pick the region with the highest upper bound makes DOO to be an optimistic algorithm in the same family as UCB (for multi-armed bandits) and HOO (for this zeroth-order setting). It then expands the leaf (h^*, i^*), evaluates f at the representative $x_{h^*+1,j}$ of the expanded regions, and updates $\mathcal{L}_t$ accordingly.

An example of DOO on $[0,1]^2$ after four epochs can be seen in Figure 1.3. Each of the regions represented is part of $\mathcal{L}_t$, and the cross are the representative (the center of the square is chosen here, but any arbitrary representative works). The algorithm should focus more on regions where f has higher value, which hints that the maximum should be on the upper right of the center of $\mathcal{X}$ (close to the currently most split cells). Yet, if this is not the case, the exploring tendency of DOO will force exploration to unvisited optimal (or close to optimal) regions of $\mathcal{X}$. As is, DOO gives no knowledge of the potential proximity of a point to the maximum, but we will alleviate this problem later while

introducing certificates.

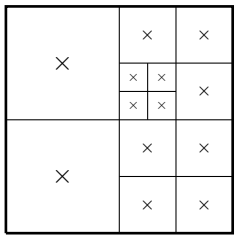


Figure 1.3: Example of the state of DOO after some epoch.

Let us now state the upper bound of the simple regret one can have using DOO, as shown in Munos (2011).[8]

Theorem 1.9. *If the near optimality dimension n_f of f is greater than zero, then the loss decreases polynomially: $R_T^s \leqslant (LR)^{n_f+1/n_f}(1-\delta^{n_f})^{-1/n_f} C_f^{1/n_f} T^{-1/n_f}$.*
If the near optimality dimension of f equals zero, then the loss decreases exponentially: $R_T^s \leqslant LR\delta^{T/C_f} - 1$.

We now describe three extensions of this setting. The first one is multi-fidelity: the learner can evaluate different approximation of the function f. The second one is certification: how to certify the output of an algorithm. The last one is f-dependent bound, where the bound given are no more worst-case, but depends on each of the function f. A setting which mixes these three settings will be presented in Chapter 4.

Multi-fidelity

In the single fidelity setting of bandits in metric spaces, we considered the black-box optimization of some function $f : \mathcal{X} \to \mathbb{R}$ that could be evaluated perfectly. We now assume that in addition to the function f it seeks to maximize, the learner can access to some approximations $\{f_\alpha | \alpha \in \Lambda\}$ of f.[9] We call α-fidelity function of f a function f_α such that for all $x \in \mathcal{X}$, $|f_\alpha(x) - f(x)| \leqslant \alpha$. Moreover, we also assume the existence of a cost function $c : \Lambda \to \mathbb{R}$ such that an evaluation of f_α at any point $x \in \mathcal{X}$ costs $c(\alpha)$. Of course, the smaller the accuracy α is, the higher the cost $c(\alpha)$ is: c is assumed to be non-increasing. The goal of the learner is to find a good trade-off between a good accuracy on the evaluation of f and a small cost incurred.

The first studied setting was with two fidelity functions: a high fidelity function f that the learner wants to minimize, and a lower fidelity function f_α for some known α which is an approximation of f, but far cheaper to evaluate. To our knowledge, the first

[8] In fact, the theorem is more general, but we only state a corollary of this theorem, to have an easier comparison with the bounds of HOO.

[9] Λ was assumed to be finite in the first papers in the literature, but soon extended to possibly be infinite in Sen et al. (2018).

researches on the subject come from the kriging community, with the study of co-kriging (Ver Hoef and Cressie, 1993). Note however that co-kriging does not explicitly focus on optimization. Years later, several papers used the low-fidelity function to explore unknown areas, and the high-fidelity to exploit the previous knowledge in areas with approximate values close to the maximum. Among many, we can quote two interesting examples. The first one (Eby et al., 1998) uses a genetic algorithm. However, instead of always using the high fidelity function, they used a genetic algorithm based on the low-fidelity function in order to populate the next generation of individuals for an iteration with the high fidelity function. This allows for a far cheaper cost for a result that can be almost as good as before. Later, Alexandrov et al. (1998) made use of this cheap low-fidelity function in a trust-region algorithm. Instead of approximating the high-fidelity by some local quadratic approximation on the trust region, it approximates it with the low-fidelity function, which provides better results as soon as the low-fidelity is closer to the high-fidelity than its second-order approximation.

From these two simple examples, more algorithms were later studied, with better theoretical foundation and empirical results.
Multiple works in the multi-fidelity setting come from the Bayesian optimization literature. For instance, Huang et al. (2006) designed a multi-fidelity variant of the EGO algorithm; Zhang et al. (2017) generalized the predictive entropy search algorithm of Hennig and Schuler (2012); Hernández-Lobato et al. (2014) and Takeno et al. (2020) generalized max-value entropy search (Wang and Jegelka, 2017); a multi-fidelity counterpart of GP-UCB (MF-GP-UCB) was studied in Kandasamy et al. (2016a, 2019), while co-kriging was studied for optimization e.g., in Forrester et al. (2007); Qian and Wu (2008). A framework where the (finitely many) fidelity functions may be mutually dependent was introduced in Song et al. (2019), alongside a high probability guarantee. A more complete review on multi-fidelity Bayesian optimization can be found in Peherstorfer et al. (2018) and in (Garnett, 2023, Section 11.5).

Recently bandits algorithms were also extended to the multi-fidelity setting. Starting with the case of finitely-many arms Kandasamy et al. (2016b), several algorithms were designed for continuously-many arms, which corresponds to multi-fidelity zeroth-order optimization *without* certificates. An extension of DOO (called MFDOO) was provided by Sen et al. (2018), together with an optimization error bound (or simple regret bound) for a given overall cost budget. The authors also study a variant inspired from POO (MFPDOO) to handle the case of unknown smoothness (see paragraph below). Similar algorithms based on HOO and POO (called MFHOO and MFPOO) were later introduced by Sen et al. (2019) to cope with additional (stochastic) noise. In Fiegel et al. (2020) the authors develop the Kometo algorithm (based on StroquOOL) and prove nearly optimal upper bounds (with matching minimax lower bounds) on the optimization error given an overall cost budget. Their analysis also covers the cases of unknown smoothness and of possibly unbounded costs at accuracies α in the neighborhood of $\alpha = 0$. The case of delayed and noisy feedback was addressed in Wang et al. (2022) with a generalization of HOO.

A typical example of a multi-fidelity real-world problem is the optimization of a func-

tion f computed with finite element modeling. A case with two fidelity functions (two values of α) appears in Sun et al. (2011), for a sheet-metal forming design with the goal of having no defects in the products (automobile inner panel in that paper). Given three variables x_1, x_2, x_3 modeling strong restraining forces on the metal, the goal is to set these forces to a good value to avoid both rupture and wrinkling. Two different finite element solvers were used to approximate f at any point x: incremental finite element solvers, or a one-step finite element model, which is computationally cheap but provides worse estimates than the former model. Finding a good design of the forces at a reasonable computational cost is an example of a two-fidelity optimization problem. Many other examples can be found, e.g., in thermodynamics Dewettinck et al. (1999); Le Gratiet (2013), design of new aircraft Geiselhart et al. (2011), or nuclear criticality safety Picheny et al. (2010). A review with further applications of multi-fidelity can be read in Fernández-Godino et al. (2016).

In chapter 4, we study a broader setting with a possibly infinite number of fidelity functions. More formally, we take $\Lambda = \mathbb{R}^*$, and $c : \mathbb{R}^* \to \mathbb{R}$ a non-decreasing function. Note that in this example, one cannot evaluate the original function f because 0 is not in Λ. In fact, most of our results can be extended to the case $\Lambda = \mathbb{R}$ as soon as c possesses a right-limit at 0, but for simplicity of redaction and better readability, we will not cover this setting. Whenever the cost c is a step function with finitely many pieces, we come back to the previous setting with a finite number of fidelities (in which case one can evaluate f at a finite cost).

A need for certification

In practice, algorithms that achieve small optimization errors are desirable but may not inform the user when such a small optimization error has been obtained. In the example above, an engineer might require to *certify* the output of the algorithm, that is, to get a guaranteed optimization error bound that they can compute by only using the observed data and some known assumptions on f, as is done, e.g., in Hansen et al. (1991) and Bachoc et al. (2021) in the single-fidelity setting. Such requirements can be important in industrial fields involving safety-critical systems (e.g., cars, aircraft, health, nuclear engineering).

Though convergence results about the optimization error $f(x_\star) - f(x_t^*)$ in terms of a total cost budget are now well established in the multi-fidelity setting, the question of *certifying* approximate maximizers with minimum total cost has not been addressed, to the best of our knowledge.

The notion of error certificate appeared in several other settings, such as, e.g., in convex optimization (Boyd and Vandenberghe, 2004), where the *duality gap* between primal and dual feasible points plays the role of an error certificate. In zeroth-order Lipschitz optimization with perfect evaluations of f (single-fidelity setting), the Piyavskii-Shubert algorithm (Piyavskii, 1972; Shubert, 1972) is naturally endowed with an error certificate, which is the difference between the maximum value of a guaranteed upper bounding func-

tion of f and the maximum value $f(x_s)$ observed so far. For one-dimensional inputs, a tight analysis of the number of evaluations before which this certificate falls below ε was given in Hansen et al. (1991) (see also Danilin (1971)), with a simple integral expression. This result was generalized to multi-dimensional inputs by Bouttier et al. (2020); Bachoc et al. (2021), and shown in Bachoc et al. (2021) to be achievable with the tractable[10] c.DOO algorithm, with a nearly matching f-dependent lower bound. In Chapter 4, we extend the complexity analysis of Bachoc et al. (2021) to the multi-fidelity setting, using a certified variant of MFDOO (Sen et al., 2018) for the upper bound.

Another important series of works is about adaptivity to the unknown smoothness of f, that is, the question of achieving nearly optimal optimization performances with an algorithm that has (almost) no prior knowledge on the smoothness of f. Among the many algorithms designed to that end, let us mention the seminal DIRECT algorithm of Jones et al. (1993), the $Z(k)$ algorithm of Horn (2006), as well as bandits algorithms (with simple or cumulative regret guarantees) including (Sto)SOO (Munos, 2011; Valko et al., 2013), the two-phase algorithm of Bubeck et al. (2011b), POO and GPO (Grill et al., 2015; Shang et al., 2019), AdaLIPO (Malherbe and Vayatis, 2017), SequOOL and StroquOOL (Bartlett et al., 2019), and their multi-fidelity variants (Sen et al., 2018, 2019; Fiegel et al., 2020). See also Locatelli and Carpentier (2018) for a detailed account on possible and impossible adaptivity results in the single-fidelity setting.

Though adaptivity to unknown smoothness is a key robustness feature of optimization algorithms, we stress that it is in a way *incompatible* with the certificate requirement. For instance, as noted by Bachoc et al. (2021) in the single-fidelity setting, when optimizing a Lipschitz function f with unknown Lipschitz constant $\mathrm{Lip}(f)$, it is impossible to produce a finite certificate ξ_t after any number t of evaluations of f, since there could be an arbitrarily steep bump in a yet unobserved input region. More formally, if f has a maximizer $x_\star$ within the interior of $\mathcal{X}$, the lower bound of (Bachoc et al., 2021, Theorem 2) scales at least as $\big(L/\mathrm{Lip}(f)\big)^n$ when $L \to +\infty$, which implies that the minimum number of evaluations that certified algorithms need for the function f is arbitrarily large, if we require such algorithms to output valid certificates for all Lipschitz functions g with arbitrarily large Lipschitz constants $\mathrm{Lip}(g)$. The same intuitive remark applies to our multi-fidelity setting, by the lower bound of Theorem 4.3 in Section 4.3.

f-dependent bounds

We have seen in Section 1.2.1 that one could provide worst-case bound and distribution-dependent bounds for the UCB algorithm. The same thing can be provided for c.MF-DOO. In the certified setting, worst-case bounds are not of much interest, as they are reached for constant functions, where c.MF-DOO and all other optimal algorithm reduce to a grid search.

Compared to the previous works that only gave f-dependent upper bound, the existence of a certificate allows us to also show f-dependent *lower bound*. Indeed, let $\mathcal{A}_x$ be the algorithm that returns a point $x \in \mathcal{X}$ at each iteration. Then, for any function f, there

[10] Tractability refers to a small (logarithmic) number of elementary operations per evaluation of f.

is one of the $\mathcal{A}_x$ that returns the maximum of f in only one iteration. Thus, without certificate, the only provable f-dependent lower bound is 1. As one can guess, it is of no practical interest. However, all of the algorithm $\mathcal{A}_x$ previously described cannot give any certificate smaller than $L \cdot \mathrm{diam}(\mathcal{X})/2$ for the point they give. This is where the inability to give appropriate f-dependent lower bounds comes, and where we see an advantage in our setting to better understand and take profit of the lower bounds.

To summarize, we consider there the problem of multi-fidelity zeroth-order optimization, where one can evaluate a function f at various approximation levels (of varying costs), and where the goal is to optimize f with the cheapest evaluations possible. In Chapter 4 , we study *certified* algorithms, which are additionally required to output a data-driven upper bound on the optimization error. We first formalize the problem in terms of a min-max game between an algorithm and an evaluation environment. We then propose a certified variant of the MFDOO algorithm and derive a bound on its cost complexity for any Lipschitz function f. We also prove an f-dependent lower bound showing that this algorithm has a near-optimal cost complexity. We close this chapter by addressing the special case of noisy (stochastic) evaluations as a direct example.

We now would like to extend certificates and provide new upper and lower bounds to the special cases of estimation of integrals and approximation of functions. This will be discussed in the next two subsections.

1.2.4 Sequential estimation of integrals

As was said at the end of Section 1.2.2, the setting of bandits with metric spaces can be changed to another goal than maximization. Its new goal is to approximate some functional $J(f)$ of the function f in study. In this section, the functional that we consider is the integral. More formally, let $\mathcal{X}$ be a compact subset of $\mathbb{R}$ and μ a probability measure on $\mathcal{X}$. For any absolutely integrable function f with respect to μ, we define $J(f)$ to be $\int_{\mathcal{X}} f(x)\,\mathrm{d}\mu(x)$.

When $\mathcal{X}$ is the segment $[0,1]$, several well-known methods can be used. Assume first that the horizon is finite and known in advance by the learner. We set it to $T+1$ for some T in $\mathbb{N}$. A first idea would be to query f on a uniform grid over $[0,1]$. Let $x_i = \frac{i}{T}$ for $i = 0, \dots, T$ be the $T+1$ points queried. Then, to approximate f on all the small sub-intervals $([x_{i-1}, x_i])_{1 \leqslant i \leqslant T}$, one could use:

- The left (resp. right) point method: approximate f by $x \mapsto f(x_{i-1})$ (resp. $x \mapsto f(x_i)$) on the segment $[x_{i-1}, x_i]$. These two methods give an error of the order of $1/T$ for functions with continuous derivatives on $[0,1]$.

- The middle point method: approximate f by $f\left(\frac{x_{i-1}+x_i}{2}\right)$, the evaluation of f at the middle point of each segment. It gives an error of the order of $1/T^2$ as soon as the function is $\mathcal{C}^2$ on $[0,1]$.

- The trapezoidal method: it approximate f by its linear interpolation

$$\widehat{f}\colon x \mapsto \frac{x - x_{i-1}}{x_i - x_{i-1}}(f(x_i) - f(x_{i-1})) + f(x_{i-1}) \,.$$

  Then the integral of $\widehat{f}$ over $[x_{i-1}, x_i]$ equals $\frac{f(x_{i-1})+f(x_i)}{2}$. As for the middle point method, it gives an error of the order of $1/T^2$ for $\mathcal{C}^2$ functions. The constants (not expressed here) are worse than the one for the middle point method, however the trapezoidal rule needs no other evaluation on each sub-interval, as compared to the middle point method.

- A last deeply used approximation working with the uniform grid is the Simpson method, which gives an error of the order of $1/T^4$ for $\mathcal{C}^4$ functions.

Another example with a non-uniform grid of points $x_0, \ldots, x_T$ is the Gaussian quadrature, an approximation based on the Legendre polynomials. Each node x_i and weights w_i (associated to $f(x_i)$) can be computed with the Golub-Welsch algorithm (Golub and Welsch, 1969). One could also use methods that do not use affine combinations of the evaluation of f to approximate its integral.

All the previous methods are non-adaptive: each point is not chosen sequentially to minimize the error, but is chosen beforehand in an offline manner (for a set horizon $T+1$). One could also think about adaptive methods that chose sequentially each query point. At a given step $t \leqslant T$, it then can use any of the previously stated method to estimate f on all of the sub-intervals $([x_{i-1}, x_i])_{1 \leqslant i \leqslant t}$. The only difference is that the sub-intervals may not be uniform. Such methods contain the previously described quadrature methods, and thus widen the field of possibilities.

For example, imagine the following toy setting where one has to estimate the integral of a piecewise constant function f with only two pieces on $[0,1]$. Let us denote by $x_\star \in [0,1]$ the unknown sole discontinuity of f. Then, using the trapezoidal rule (uniform grid with affine interpolation), the worst-case bound is of the order of $1/T$. However, using a sequential method with a dichotomy to find $x_\star$ and then an affine interpolation gives a worst-case bound of the order of $\exp(-T)$, faster by a lot than the previous method.

From this example, one could think that adaptive methods are always better than non-adaptive ones, because they give more flexibility, and can adapt to each function (as their name tells us). However, it was proven that in some settings, they do not improve the worst-case estimation error!
Indeed, let us denote by $\mathcal{F}$ the set of functions studied, (for example, the set of integrable functions from $\mathcal{X}$ to $\mathbb{R}$ in the first example). Then, Bakhvalov (1971) showed that if the set $\mathcal{F}$ is convex and symmetric, adaptive methods are not better than non-adaptive ones. In the same vein, Sukharev (1986) proved that for any convex set $\mathcal{F}$, arbitrary non-adaptive methods are not better than affine ones. This is why, in many settings, while simple, the methods given previously provide minimal estimation error of the integral.

Another extension of the previous methods should be spoken about: randomized methods. The randomization may appear both in the choice of the query points or in the approximation of f on each interval. Here, we focus on the second kind of randomization. An example of such an algorithm is the randomized trapezoidal rule (Wu, 2022). In that paper, the author shows that the convergence of the trapezoidal rule was improved by a factor $T^{1/2}$ with randomization for any function in Sobolev spaces. However, improvement with randomization does not work for every class of functions.
Now, let us study the specific case of monotone functions, and see how we can apply the previous literature to this setting, and if it is possible to provide new bounds.

1.2.5 Approximation and integral estimation of monotone functions

In Chapter 5, we study specifically *monotone functions*. We study integral estimation of monotone functions in Section 5.4.1, with the introduction and theoretical study of GreedyBox, an optimal algorithm (up to logarithmic factors) greatly inspired by the work of Novak (1992). We also improve in expectation the estimation error to $T^{3/2}$ with a stochastic version of this algorithm, which achieves the worst-case lower bound of adaptive randomized methods proved by Novak (1992).

Note that the set $\mathcal{F}$ of monotone functions on a segment is convex but not symmetric. Thus, according to the results presented in the previous section, we know that arbitrary non-adaptive methods are not better than affine ones on this set. However, Kiefer (1957) later showed that the trapezoidal rule is optimal among all deterministic (possibly adaptive) methods for integral approximation in the case of monotone functions. His work was completed by that of Novak (1992), who gave optimal bounds for different possible types of algorithms as summarized in Table 1.1. In particular it is shown that adaption combined with randomization is key to obtaining an improved rate in expectation. These bounds were later extended by Papageorgiou (1993) to the integral approximation of multivariate monotone functions. The books from Davis and Rabinowitz (2007) and Brass and Petras (2011) give a larger panel of results on numerical integration under various assumptions.

Strategy	**Non-adaptive**	**Adaptive**
Deterministic	ε^{-1}	ε^{-1}
Stochastic	ε^{-1}	$\varepsilon^{-2/3}$

Table 1.1: Minimax rates proved by Novak (1992).

Of course, many other function sets were studied. A non-exhaustive list includes work on the set of unimodal functions (Novak and Roschmann, 1996), on the set of functions with bounded variation (Graf and Novak, 1990), on convex and symmetric classes of functions (Novak, 1993, 1995; Hinrichs et al., 2011), and on various classes of multivariate functions (Ritter et al., 1993; Katscher et al., 1996; Krieg and Novak, 2017). Note that the bounds in the previous papers are not f-dependent.

From integral estimation to approximation: All the previous works were on integral approximation, an easier problem than L^p approximation. Many works study the L^p approximation of diverse classes of functions, mostly using interpolation. A known example is polynomial interpolation for k-times continuously differentiable functions. Many works employ the modulus of smoothness of the function f to bound its L^p approximation, providing f-dependent bounds. It is for example the case of Chandra (2002) for the approximation of periodic functions using trigonometric polynomials. The computation of a best linear L^p-approximation, where a basis $\phi_1, \dots, \phi_k$ of functions is previously given and one looks for the weights $w \in \mathbb{R}^k$ that minimize $\|f - \sum_i w_i \phi_i\|_p$, was studied in depth (e.g. (Fletcher et al., 1971, 1974)). Plaskota and Wasilkowski (2005) studied the class of functions with their first k derivatives continuous except at one singularity, and showed that adaptive algorithms are better that non-adaptive in the case of integral estimation. The same remark and work was later carried to approximation in Plaskota et al. (2008). Some work involves k-monotone functions, that is, functions with monotone k-th derivatives. The papers Kopotun and Shadrin (2003); Kopotun et al. (2009) studied the rate of convergence of interpolation methods on this set of functions, and showed results depending on the modulus of smoothness of the function. To our knowledge, little is known about $L^p(\mu)$ approximation of general non-decreasing functions, without any continuity or smoothness assumptions.

Adaptive methods have a long history in numerical integration and other approximation or learning problems. For numerical integration of monotone functions, as mentioned above, adaption combined with randomization is key to improving worst-case rates (see Novak (1992) and Table 1.1 above, as well as Novak (1996)). We show that adaption is also key to obtaining less pessimistic, f-dependent error bounds. This work also shares algorithmic principles with online learning methods for (possibly noisy) black-box optimization, such as bandit algorithms (Lattimore and Szepesvári, 2020) or the EGO algorithm (Jones et al., 1998). Indeed such algorithms rely on adaptive sampling strategies to reduce the current uncertainty (via, e.g. UCB or Bayesian approaches), which is reminiscent of the way GreedyBox selects the box to be split at time t. Close to Chapter 5 is the work by Bachoc et al. (2021), who derive f-dependent error bounds for certified black-box Lipschitz optimization.

In Sections 5.2 and 5.3, we study the problem of $L^p(\mu)$ approximation for monotone functions. Given a monotone function f, we do not want to find an approximation of $J(f)$ anymore, but an approximation of f itself, where the error is computed with the $L^p(\mu)$ norm defined by $\|f - g\|_p = \left(\int_{x \in \mathcal{X}} |f(x) - g(x)|^p \, \mathrm{d}\mu(x)\right)^{\frac{1}{p}}$. When $p = 1$, to have a good approximation of f implies to have a good estimation of the integral. Keeping GreedyBox provides the same bound as for integral estimation when $p = 1$, and the bound found can be generalized to any $p \geqslant 1$. The upper bounds we provide are f-dependent, and match up to logarithmic factors the according lower bounds.

Chapter 2

Introduction en Français

La confiance a toujours été la pierre angulaire des interactions entre être humains, car au cœur de chaque homme se trouve une quête de sens et de vérité. Ce principe fondamental s'applique universellement, étendant son influence même dans le domaine des mathématiques. De l'encadrement de π par Archimède il y a plus de deux millénaires jusqu'aux dernières mesures de précision des constantes fondamentales de notre univers apportées par les résultats de Planck 2018 (Aghanim et al., 2020), les mathématiciens et physiciens cherchent toujours à fournir des garanties plus strictes et plus fiables.

Dans le monde actuel, où les algorithmes sont présents dans tous les aspects de notre vie, les algorithmes certifiables jouent un rôle central dans l'élargissement de cette confiance et de ces garanties aux nouvelles technologies. Obtenir des algorithmes qui convergent plus rapidement est d'un faible intérêt si l'on ne sait pas s'ils convergent vers un bon résultat. De même, des algorithmes précis avec une bonne vitesse de convergence devraient obtenir des résultats plus fiables pour être utilisés à grande échelle dans les logiciels actuels. De meilleures explications et une certifiabilité plus approfondie sont nécessaires. Dans des domaines critiques tels que les systèmes nucléaires, l'innovation technologique ou la santé, existe une forte incitation pour une meilleure compréhension des algorithmes utilisés couplée à la recherche de bornes de sécurité sur ceux-ci.

Dans cette thèse, nous abordons ces défis à travers trois algorithmes différents. Nous fournissons d'abord des bornes théoriques sur un algorithme utilisé en théorie des jeux, un domaine des mathématiques étroitement lié à l'économie et à l'apprentissage automatique. Nous nous occupons ensuite de l'ajout de certificats et de l'amélioration de certaines bornes sur les vitesses de convergence pour de l'optimisation avec des évaluations approximatives, suivant ainsi la voie de nombreux algorithmes de l'optimisation globale, de l'optimisation bayésienne et des problèmes de bandits en espaces métriques. L'algorithme étudié peut être d'une grande utilité dans de nombreux problèmes d'ingénierie. En dernier lieu, nous étudions le problème de l'approximation de fonction et, une fois encore, proposons un algorithme avec certificat.

À travers ce travail, nous visons à contribuer à la quête toujours changeante de la

confiance, de la précision et de la sécurité dans les mathématiques et au-delà.

Contents

2.1 De l'optimisme pour apprendre dans les jeux 27
2.1.1 Des jeux à somme nulle aux jeux à somme générale 27
2.1.2 Équilibre de Nash 29
2.1.3 Apprentissage en théorie des jeux 30
2.1.4 Méthodes pour l'apprentissage et l'évolution dans les jeux 31
2.1.5 Un besoin d'optimisme pour apprendre 33
2.1.6 Articles et études antérieures 34
2.2 Bandits à valeur dans un espace métrique pour l'optimisation et l'approximation 35
2.2.1 Bandits contre un adversaire 35
2.2.2 Bandits dans un espace métrique 39
2.2.3 Méthodes pour des bandits à valeur dans un espace métrique . . 40
Multi-fidelité 44
De la nécessité d'un certificat 47
2.2.4 Estimation séquentielle d'intégrales 48
2.2.5 Approximation et estimation d'intégrales de fonctions monotones 50

2.1 De l'optimisme pour apprendre dans les jeux

2.1.1 Des jeux à somme nulle aux jeux à somme générale

Les jeux à somme nulle sont des jeux entre deux joueurs (Joueur 1 et Joueur 2), où les deux jours possèdent des intérêts différents. Dans les jeux à somme nulle, ce que le deuxième joueur perd est gagné par le premier. Des jeux à un tour comme Pierre-papier-ciseaux ou pile ou face en sont un bon exemple. D'autres jeux peuvent être sous forme d'arbre de décision en plusieurs tours. Cependant, ces derniers peuvent toujours être réduits à des jeux à un seul coup. C'est par exemple le cas du morpion ou des échecs.

Commençons par introduire les jeux finis à somme nulle. Pour de tels jeux, chaque joueur a un nombre fini d'actions (ou de stratégies). Nous désignons par I (de cardinal n) l'ensemble des actions du Joueur 1, et par J (de cardinal m) l'ensemble des actions du Joueur 2. À chaque couple $(i, j) \in I \times J$ d'actions potentielles des deux joueurs est associée une récompense $a_{i,j}$. Le Joueur 2 doit payer le montant $a_{i,j}$ au Joueur 1. Dans l'exemple du jeu Pierre-papier-ciseaux, si le Joueur 1 joue "papier" (action 2) et que le Joueur 2 joue "pierre" (action 1), alors le deuxième joueur doit payer $a_{2,1} = 1$ au premier joueur. Plus le montant est élevé, plus le Joueur 1 sera content, au détriment de l'autre joueur qui tente de minimiser sa perte. Notons que $a_{i,j}$ peut être négatif, auquel cas le Joueur 2 y gagne, en donnant un paiement négatif au Joueur 1. Les deux joueurs doivent jouer simultanément et ne peuvent donc pas jouer en fonction de l'action de l'autre. Ils devraient donc se préparer à jouer contre n'importe quelle action possible de l'autre joueur.

Parce que le jeu est uniquement déterminé par ses récompenses $(a_{i,j})_{(i,j)\in I\times J}$, il peut être représenté par une matrice A. C'est pourquoi, les jeux finis à somme nulle sont également appelés jeux matriciels. Par exemple, la matrice associée au jeu Pierre-papier-ciseaux est la matrice

$$A_{\text{PPC}} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}.$$

Remarquons que si l'on note e_i le i-ème vecteur de la base canonique, alors $a_{i,j} = e_i^T A e_j$. On peut alors se demander si les actions de chaque joueur peuvent être étendues à n'importe quel vecteur (x, y) dans $\mathbb{R}^n \times \mathbb{R}^m$. Une extension naturelle consiste à jouer de manière aléatoire entre différentes actions. Par exemple, supposons que le Joueur 1 veuille jouer "pierre" avec une probabilité $\frac{1}{2}$ et "feuille" avec une probabilité $\frac{1}{2}$. Dans ce cas, son action est $\frac{\delta_{e_1}+\delta_{e_2}}{2}$ où δ_x désigne la mesure de Dirac en un certain point $x \in \mathbb{R}^n$. En moyenne, cette action vaut $\mathbb{E}\left[\frac{\delta_{e_1}+\delta_{e_2}}{2}\right] = \frac{e_1}{2} + \frac{e_2}{2} = \left(\frac{1}{2}, \frac{1}{2}\right)$, et son gain moyen contre une action y du deuxième joueur est $\frac{e_1}{2} + \frac{e_2}{2} = \left(\frac{1}{2}, \frac{1}{2}\right)^T Ay$. Nous disons qu'une telle action est une stratégie mixte, au contraire d'une action non aléatoire e_i, appelée stratégie pure. Par abus de notation, nous définissons une stratégie mixte par son action moyenne et son gain par ce qu'elle gagne en moyenne. Ainsi, les actions mixtes se trouvent dans le n-simplexe (respectivement m-simplexe pour le Joueur 2) défini par $\Delta(n) = \{x \in \mathbb{R}^n_{\geqslant 0} : \sum_{i=1}^n x_i = 1\}$. Par habitude, on note par $x \in \Delta(n)$ et $y \in \Delta(m)$ les stratégies mixtes du Joueur 1 et du Joueur 2 respectivement. Selon la définition précédente, le gain pour un couple (x, y) de stratégies mixtes est donc le scalaire $x^T Ay$. Les jeux ayant leurs stratégies dans le simplexe sont appelés jeux contraints.

Une autre extension consiste à permettre de jouer n'importe quelle action de l'ensemble $\mathbb{R}^n \times \mathbb{R}^m$. Une fois encore, après avoir joué (x, y), le gain est $x^T Ay$. De tels jeux sont appelés jeux non contraints. Les jeux contraints sont plus populaires et présentent des propriétés de robustesse en plus.

Pour l'instant, tous les jeux présentés sont bilinéaires, mais nous pouvons envisager des jeux avec des gains plus complexes. Par exemple, nous pouvons considérer des gains concaves/convexes (concaves par rapport à la première variable et convexes par rapport à la deuxième) sur des ensembles compacts et convexes $\mathcal{X}$ et $\mathcal{Y}$. Pour l'instant, nous revenons au cadre plus classique des jeux à somme nulle avec gains mixtes, en raison de leurs propriétés de robustesse, pour énoncer plus clairement les définitions qui nous seront utiles par la suite. Nous reviendrons plus tard aux jeux non contraints et concaves/convexes.

Une autre généralisation possible est l'utilisation de jeux à somme non nulle, aussi appelés *jeux à somme générale*. Dans ce cas, ce que le Joueur 1 gagne n'est pas nécessairement ce que le Joueur 2 perd. Imaginons par exemple un scénario où deux amis aimeraient faire une activité ensemble. Le premier souhaite voir un match de football, tandis que le second préférerait aller à un match de rugby. Bien sûr, qu'importe le match, ils préféreraient y aller ensemble. On peut représenter cette situation avec le tableau suivant :

$$\begin{pmatrix} (2,1) & (0,0) \\ (0,0) & (1,2) \end{pmatrix}$$

où pour chaque couple d'actions (i,j), deux coefficients $a_{i,j}$ et $b_{i,j}$ sont reçus, l'un étant le gain du premier ami et l'autre le gain du second ami. Ce jeu s'appelle "La bataille des sexes". Ce problème peut ainsi être modélisé par deux matrices A et B de taille $n \times m$. Un jeu à somme générale est donc désigné par ses matrices de gains (A,B), ou plus généralement par ses gains (g_1,g_2) pour les jeux non matriciels.

2.1.2 Équilibre de Nash

Dans le cas de Pierre-papier-ciseaux, si un joueur joue toujours "pierre", l'autre joueur doit jouer "feuille" pour gagner. Cela correspond à la notion de meilleure réponse : la meilleure action possible contre une action spécifique de l'autre joueur. Une meilleure réponse n'est pas toujours unique. Par exemple, dans le jeu à somme nulle $\begin{pmatrix} 1 & 3 \\ 1 & -3 \end{pmatrix}$, si le Joueur 2 joue l'action 1, les deux actions 1 et 2 sont des meilleures réponses du Joueur 1, donnant un gain de 1 dans les deux cas. Nous notons par $\mathrm{BR}_1(y)$ (resp. $\mathrm{BR}_2(x)$) l'ensemble des meilleures réponses pour le Joueur 1 (resp. le Joueur 2) contre une action x du Joueur 2 (resp. y du Joueur 1). Formellement, l'ensemble des meilleures réponses est défini par $\mathrm{BR}_1(y) = \mathrm{Arg}\max_{x\in\Delta_n} x^T A y$ et $\mathrm{BR}_2(x) = \mathrm{Arg}\max_{y\in\Delta_m} x^T B y$.

Nous appelons équilibre de Nash tout couple $(x_\star, y_\star) \in \Delta(n) \times \Delta(m)$ de stratégies tel que x est une meilleure réponse contre y et y est une meilleure réponse contre x. Alors, pour tout $x \in \Delta(n)$, $y \in \Delta(m)$, $x^T A y_\star \leqslant x_\star^T A y_\star$ et $x_\star^T B y_\star \leqslant x_\star^T B y$. Dans le cas des jeux à somme nulle, cela signifie que $(x_\star, y_\star)$ est un point selle de la fonction de gain.

Pour les jeux à un seul tour, ce dont on a vraiment besoin n'est pas d'être bon contre l'autre joueur, car on ne connaît pas sa stratégie à l'avance, mais d'être bon contre n'importe quelle stratégie possible du joueur. Pour le Joueur 1, cela signifie trouver une stratégie $x_\star$ qui maximise $\max_{x\in\Delta(n)} \min_{y\in\Delta(m)} x^T A y$. Une telle stratégie est appelée optimale pour le Joueur 1.

Un résultat important en théorie des jeux est le théorème du minimax de von Neumann pour les jeux de deux joueurs à somme nulle (von Neumann, 1928), étendu plus ultérieurement par Nash à n'importe quel jeu fini à n joueurs (Nash, 1951) :

Théorème 2.1 (Théorème minimax de von Neumann)**.** *Soit $A \in \mathbb{R}^{n\times m}$ une matrice de gains. Alors*

$$\max_{x\in\Delta(n)} \min_{y\in\Delta(m)} x^T A y = \min_{y\in\Delta(m)} \max_{x\in\Delta(n)} x^T A y$$

Nous appelons valeur du jeu A, notée $Val(A)$, cette quantité.
De plus, les deux joueurs ont une stratégie optimale.

Notez qu'une généralisation existe pour le cas plus général des ensembles concaves/convexes (Sion, 1958) :

Théorème 2.2 (Théorème de Sion)**.** *Soient $\mathcal{X}$ et $\mathcal{Y}$ deux ensembles compacts convexes non vides, et $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ une fonction continue et concave/convexe. Alors le jeu possède une valeur :*

$$\max_{x\in\mathcal{X}} \min_{y\in\mathcal{Y}} g(x,y) = \min_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} g(x,y)$$

De plus, chacun des joueurs possède une stratégie optimale.

Nous avons également mentionné les jeux bilinéaires non contraints, c'est-à-dire les jeux ayant comme ensemble d'actions tout l'espace euclidien. Étant donné que la preuve des théorèmes de Nash et de Sion repose sur la compacité des ensembles, on pourrait se demander si l'existence d'un équilibre de Nash et d'une valeur existe pour de tels jeux. La réponse est affirmative, et découle de résultats de base d'algèbre linéaire. Cela peut même être étendu aux jeux à somme générale.

Lemme 2.3. *Soit $A \in \mathbb{R}^{n\times m}$ une matrice de gains d'un jeu non contraint. Alors le jeu possède une valeur valant 0, et le jeu a au moins un équilibre de Nash $(x_\star, y_\star) \in \mathbb{R}^n \times \mathbb{R}^m$.*

Par la suite, nous désignons par $\mathrm{Ker}(A)$ le noyau d'une matrice $A \in \mathbb{R}^{n\times m}$, c'est-à-dire l'ensemble de tous les vecteurs $y \in \mathbb{R}^m$ tels que $Ay = 0$. Nous désignons également par $\mathrm{Im}(A)$ l'image de A, c'est-à-dire l'ensemble de tous les $z \in \mathbb{R}^n$ tels qu'il existe un $y \in \mathbb{R}^n$ avec $z = Ay$.

Proof. Soit $(x_\star, y_\star)$ dans $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$. Alors pour tous $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$,

$$x^T A y_\star \leqslant x_\star^T A y^\star \leqslant x_\star^T A y,$$

tous ces termes valant 0. □

Le théorème précédent est toujours valable pour les jeux bilinéaires à somme générale, puisque la valeur d'un jeu n'a pas de signification dans ce cas. Cependant, dans ce cadre, il n'est plus toujours bénéfique pour les joueurs de jouer un équilibre de Nash. Par exemple, considérez le cas où $A = B = I_n$, où I_n est la matrice identité de dimension $n = m$ avec des uns sur la diagonale et des zéros partout ailleurs. Dans ce cas, il est toujours bénéfique pour les joueurs de jouer le même vecteur avec la norme la plus élevée possible. Leur gain peut être simultanément arbitrairement grand. Ce comportement sera observé ultérieurement dans le chapitre 3, et est similaire à l'équilibre corrélé grossier (Coarse Correlated Equilibrium).

2.1.3 Apprentissage en théorie des jeux

Dans la réalité, de nombreux jeux sont joués de manière séquentielle. Les parties de Pierre-papier-ciseaux sont très souvent des jeux en plusieurs manches. Le gagnant est alors le concurrent qui remporte le plus de manches après un nombre préalablement fixé de tours. Dans ce contexte, nous aimerions savoir comment la stratégie d'un joueur évolue au cours du jeu. *L'apprentissage en théorie des jeux* est l'étude de la manière dont les joueurs apprennent à jouer à un jeu, de l'action ou de l'ensemble d'actions vers lesquelles la stratégie de chaque joueur converge, ainsi que de la vitesse de convergence. À partir de maintenant, nous considérons des jeux avec un nombre infini d'étapes.

Soit $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ une fonction de paiement. Le protocole est le suivant : En même temps, le Joueur 1 choisit une action $x_1 \in \mathcal{X}$ et le Joueur 2 choisit une action $y_1 \in \mathcal{Y}$. Ensuite, le Joueur 2 paie $g(x_1, y_1)$ au Joueur 1, et chaque joueur observe l'action jouée par son adversaire. Maintenant, à chaque instant t, les deux joueurs ont accès à l'historique

entier $h_{t-1} = (A^T x_1, Ay_1, \dots, A^T x_{t-1}, Ay_{t-1})$ des précédents gradients observés, et choisissent leur prochaine action $x_t \in \mathcal{X}$ et $y_t \in \mathcal{Y}$ en conséquence. Ils reçoivent ensuite le paiement $g(x_t, y_t)$. Nous désignons par $\sigma = (\sigma_t)_{t \geqslant 1}$ une stratégie pour le Joueur 1, et par τ une stratégie pour le Joueur 2. Pour tout $t \geqslant 1$, σ_t est une fonction qui associe à toute valeur possible $h_{t-1} \in (\mathbb{R}^n \times \mathbb{R}^m)^{t-1}$ de l'historique une action $x_t \in \Delta(n)$.

2.1.4 Méthodes pour l'apprentissage et l'évolution dans les jeux

Dans cette section, nous étudions les jeux avec de multiples tours, et en particulier les méthodes que les joueurs peuvent utiliser pour converger vers un équilibre de Nash du jeu. La première méthode à laquelle nous pourrions penser est de jouer la meilleure réponse contre la moyenne empirique de son adversaire. C'est-à-dire, à chaque instant t, la stratégie σ_t du Joueur 1 est de jouer n'importe quel x_{t+1} dans $\mathrm{BR}_1\left(\frac{1}{t-1}\sum_{s=1}^{t-1} y_s\right)$. Cet algorithme, introduit dans Brown (1951), est appelé Fictitious Play. Notons que la meilleure réponse peut ne pas être définie dans le cas non contraint. Cependant, Fictitious Play est généralement étudié pour des stratégies mixtes. On sait qu'il converge en moyenne vers un équilibre de Nash pour les jeux à somme nulle à deux joueurs (Robinson, 1951). Néanmoins, la vitesse de convergence est polynomiale en fonction de t, et peut être très lente à mesure que le nombre d'actions augmente (Daskalakis and Pan, 2014), contrairement à la conjecture faite dans (Karlin, 2003, Section 6.6). De plus, pour Fictitious Play *la suite des derniers itérés ne converge pas*, du moins pour les jeux ne possédant que des équilibres de Nash mixtes au moins pour les jeux sans équilibre de Nash pur, car alors les actions de chaque joueur alternent entre différentes stratégies pures. Il existe une version stochastique de Fictitious Play introduite dans (Fudenberg and Kreps, 1993) dont on sait que *les derniers itérés convergent* depuis le travail de Hofbauer and Sandholm (2002).

Divers algorithmes ont été proposés pour trouver un point selle d'un jeu. En voici une liste non exhaustive : Arrow et al. (1958); Rockafellar (1976); Nemirovski and Yudin (1983); Freund and Schapire (1999). Nous nous concentrons ici sur une variante de la montée-descente de gradient (dont l'acronyme anglais est GDA, représenté par l'algorithme 5) appelée montée-descente de gradient optimiste (notée OGDA, pour Optimistic Gradient Descent Ascent en anglais), dans le cas d'un pas de gradient $\eta > 0$ constant.

Algorithm 5 Montée-descente de gradient

Entrée: Une fonction g
Initialisation: Une taille de pas $\eta > 0$, une initialisation $(x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$
pour $t = 0, \dots,$ **faire**
$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t + \eta \frac{\partial g}{\partial x}(x_t, y_t)\right)$
$y_{t+1} = \Pi_{\mathcal{Y}}\left(y_t - \eta \frac{\partial g}{\partial y}(x_t, y_t)\right)$
fin pour

Tout d'abord, rappelons l'algorithme de base qu'est GDA, pour lequel à chaque itération t, les deux joueurs effectuent un pas de gradient. Les réseaux antagonistes génératifs

(abrégé en GAN, venant de l'anglais Generative Adversarial Networks, Goodfellow et al. (2014); Biau et al. (2020)), sont un bon exemple de l'utilisation de cet algorithme. Ils consistent en deux réseaux de neurones profonds qui jouent l'un contre l'autre dans le but, pour le premier d'approcher une distribution de probabilité, et pour le deuxième de séparer les vraies données des données générées. La plupart des GANs actuels utilisent (une version stochastique) de la montée-descente de gradient. Dans plusieurs jeux à somme nulle, il a été montré que GDA converge en moyenne vers un équilibre (Nedić and Ozdaglar, 2009). Cependant, il est bien connu que même pour le problème simple $g(x,y) = xy$ avec $\mathcal{X} = \mathcal{Y} = \mathbb{R}$, GDA peut présenter un comportement cyclique et son *dernier itéré peut diverger*.

Proposition 2.4. *Soit $A \in \mathbb{R}^{n\times p}$ une matrice de paiement d'un jeu bilinéaire non contraint à somme nulle. Alors, la dynamique de GDA peut ne pas converger vers un équilibre de Nash du jeu de matrice de gain A.*

Proof. Pour le jeu de l'appariement des sous, où $n = m = 1$ et $A = (1)$, $x^T Ay$ est tout simplement égal à xy. Dans ce cas, nous obtenons

$$x_{t+1} = x_t + \eta y_t \text{ et } y_{t+1} = y_t - \eta x_t$$

ce qui implique que $x_{t+1}^2 + y_{t+1}^2 = (x_t^2 + y_t^2)(1+\eta^2)$. Cela signifie que $\|(x_t, y_t)\| \xrightarrow[t\to\infty]{} +\infty$ dès que $(x_0, y_0) \neq (0,0)$.

□

La proposition 2.4 ainsi qu'un résultat de convergence du dernier itéré peuvent être étendus à la montée-descente miroir (MDA, voir Shalev-Shwartz and Singer (2006)), une adaptation de l'algorithme de descente miroir de Nemirovski and Yudin (1983) à un problème minimax. L'algorithme de descente miroir met à jour la stratégie avec une étape de gradient dans un "espace miroir" qui dépend d'une fonction de régularisation (une fonction fortement convexe). Lorsque la fonction de régularisation est la norme euclidienne, MDA se réduit à GDA, et lorsque la fonction de régularisation est l'entropie négative sur le simplexe, on obtient l'algorithme des poids exponentiels.
Une bonne variante de MDA pour surmonter le comportement cyclique et divergent de la proposition 2.4 est l'algorithme de montée-descente miroir optimiste. Il se réduit à l'algorithme de montée-descente de gradient optimiste (OGDA, algorithme 6 quand la fonction de régularisation utilisée est la norme euclidienne.

Algorithm 6 Descente-montée de gradient optimiste (OGDA)

Entrée: Une fonction g
Initialisation: Un pas de taille $\eta > 0$, une initialisation $(x_0, y_0, x_{-1}, y_{-1}) \in \mathbb{R}^{(n+m+n+m)}$
pour $t = 0, \ldots,$ **faire**
$x_{t+1} = \Pi_{\mathcal{X}} \left(x_t + 2\eta \frac{\partial g}{\partial x}(x_t, y_t) - \eta \frac{\partial g}{\partial x}(x_{t-1}, y_{t-1}) \right)$
$y_{t+1} = \Pi_{\mathcal{Y}} \left(y_t - 2\eta \frac{\partial g}{\partial y}(x_t, y_t) + \eta \frac{\partial g}{\partial y}(x_{t-1}, y_{t-1}) \right)$
fin pour

Un exemple du comportement cyclique de l'algorithme de montée-descente de gradient et de la convergence de son homologue optimiste peut être observé sur la Figure 2.1 pour le jeu de l'appariement des sous, défini par la matrice de gain $A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. Pour les stratégies mixtes, l'équilibre de Nash du jeu consiste à jouer la distribution uniforme sur l'ensemble des actions possibles pour les deux joueurs.

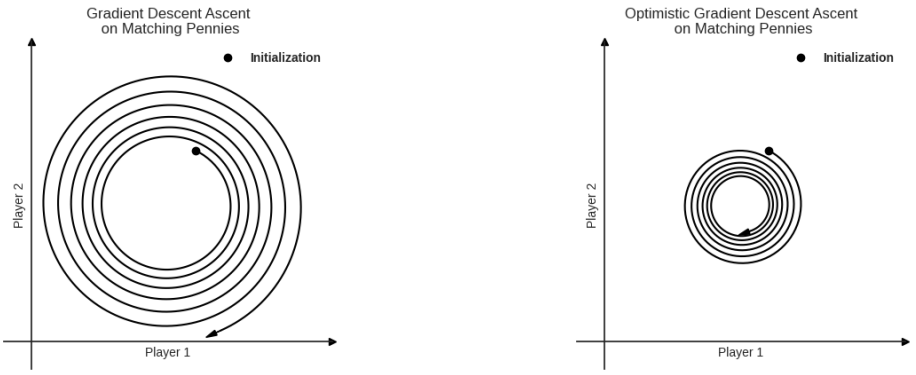


Figure 2.1: Comparaison de GDA et OGDA pour le jeu des "Matching Pennies"

2.1.5 Un besoin d'optimisme pour apprendre

Dans la suite, nous considérons le cas bilinéaire sans contrainte, où $\mathcal{X}$ et $\mathcal{Y}$ sont des espaces euclidiens, c'est-à-dire qu'il existe des entiers $n, m \geqslant 1$ tels que $\mathcal{X} = \mathbb{R}^n$ et $\mathcal{Y} = \mathbb{R}^m$. Pour cette raison, nous utilisons la norme euclidienne comme fonction de régularisation, et l'algorithme que nous étudions est OGDA. Nous nous en tenons ici au cas standard où les poids des gradients de l'étape $t-1$ sont égaux à la moitié des poids des gradients de l'étape t, mais d'autres choix sont possibles. On peut consulter le travail de Peng et al. (2020) pour observer le comportement pour d'autres rapports entre les coefficients.

Cet algorithme *optimiste* a été introduit par Korpelevich (1976); Popov (1980) et redécouvert dans le cas des GANs des décennies plus tard (Daskalakis et al., 2018). Un défi consiste alors à déterminer à quel moment OGDA converge vers un équilibre. En théorie des jeux, un algorithme est dit optimiste lorsqu'il utilise une étape intermédiaire pour calculer le gradient : le point au temps $t+1$ est calculé à l'aide du gradient au temps t (comme dans les algorithmes classiques jusqu'à présent) *et* du gradient à un temps $t+1/2$.[1] L'algorithme de montée-descente de gradient optimiste est un exemple d'un tel algorithme optimiste avec un seul appel à la fonction par tour : nous ne calculons pas le gradient à un instant $t+1/2$, mais nous utilisons les gradients précédemment calculés pour l'approcher.

[1] Attention : cette notion d'optimisme est différente de celle utilisée dans le contexte des bandits. Cette deuxième forme d'optimisme sera abordée dans la section 2.2.

Ici, nous le faisons en utilisant le gradient au temps $t-1$.

Dans le Chapitre 3, nous étudions la convergence d'OGDA pour les jeux bilinéaires non contraints. Nous fournissons le meilleur ratio de convergence en fonction du pas de gradient η, et trouvons le pas de gradient donnant la convergence la plus rapide. Ensuite, une étude d'OGDA pour les jeux à somme générale est effectuée. Nous y montrons que OGDA converge vers des actions appropriées pour un grand ensemble de jeux. Cette convergence est ensuite utilisée pour accélérer la convergence dans un problème à somme nulle en introduisant OGDA sur un jeu à somme générale bien choisi. Cela montre de manière intéressante que les jeux à somme générale peuvent être utilisés pour améliorer les algorithmes conçus pour les jeux à somme nulle. Nous examinons ensuite des résultats expérimentaux, que ce soit dans le cadre bilinéaire ou dans le cadre concave/convexe.

2.1.6 Articles et études antérieures

La version optimiste de l'algorithme de GDA a été étudiée pour la première fois dans un article de Popov (1980). Rakhlin and Sridharan (2013) ont été les premiers à introduire la montée-descente miroir optimiste, et ont montré que cet algorithme est sans regret.

Dans le contexte des GANs, une première étude d'OGDA pour les jeux bilinéaires a été menée par Daskalakis et al. (2018). Ils ont démontré la convergence du dernier itéré d'OGDA, dans le sens où, pour chaque $\varepsilon > 0$, on peut trouver $\eta > 0$ et T suffisamment grand pour que (x_T, y_T) (tel que défini dans l'Algorithme 6) soit à ε près d'un équilibre de Nash. Plus tard, Liang and Stokes (2019) ont prouvé que dans le cas d'une matrice carrée inversible A, OGDA converge vers $(0,0)$ à une vitesse exponentielle pour une valeur spécifique du pas η. Gidel et al. (2019) ont étudié les GANs à travers des inégalités variationnelles, et ont donné des bornes plus faibles mais plus générales que celle de Liang and Stokes (2019) pour la vitesse de convergence.

Une autre variante de GDA est Extra-Gradient (EG), une méthode introduite par Korpelevich (1976). Mokhtari et al. (2020a) ont montré que OGDA et EG peuvent être vus comme des approximations de la méthode du point proximal, et ont prouvé la convergence exponentielle de ces trois algorithmes pour les jeux bilinéaires carrés inversibles et une valeur particulière de η.
Zhang and Yu (2020) ont étudié en profondeur GDA, EG, ODGA ainsi qu'une autre méthode basée sur les moments. Ils ont notamment trouvé le pas optimal $\eta^\star$ et son taux géométrique associé $\lambda^\star$ pour OGDA en fonction de la matrice A. Un autre travail en lien avec tout ceci est celui de Hsieh et al. (2019) qui ont étudié, entre autres choses, le cadre stochastique pour EG et OGDA. Les auteurs ont également mis en évidence le lien étroit qui réunit ces deux algorithmes.

Au-delà des jeux bilinéaires, plusieurs articles ont examiné OGDA pour les jeux concaves/convexes. En particulier, Daskalakis and Panageas (2018) ont étudié la stabilité des points fixes de la dynamique d'OGDA, et Abernethy et al. (2018) ont amélioré le taux de convergence pour certains jeux concaves/convexes spécifiques. Un article ultérieur de Lin et al. (2020) a atteint, pour un autre algorithme de leur création, la borne inférieure

pour les jeux fortement concaves/fortement convexes de Ibrahim et al. (2020) (par la suite généralisée dans Hadiji et al. (2023)), à des facteurs logarithmiques près. Cependant, le cadre non concave/non convexe présente des propriétés problématiques : Hsieh et al. (2021) ont montré qu'il peut exister des attracteurs d'OGDA contenant aucun point stationnaire, et Daskalakis et al. (2021) ont montré que trouver un équilibre local min-max approximatif est PPAD-dur. Néanmoins, dans un article récent, Diakonikolas et al. (2021) ont montré des résultats positifs dans une nouvelle classe de problèmes min-max non concaves/non convexes.

De nombreuses études ont également été menées dans des contextes compacts et convexes, c'est-à-dire lorsque $\mathcal{X}$ et $\mathcal{Y}$ sont des ensembles compacts. Daskalakis and Panageas (2019) ont prouvé la convergence d'une version optimiste des Multiplicative Weight Updates sur le simplexe, pour une valeur appropriée du taux d'apprentissage. Plus tard, Wei et al. (2021) ont amélioré leurs résultats en montrant la convergence pour n'importe quel taux d'apprentissage inférieur à une constante. Enfin, dans un article très récent, Anagnostides et al. (2022) ont considéré la convergence d'une version projetée d'OGDA pour les jeux de somme générale, où après chaque itération, une projection sur le simplexe est appliquée.

2.2 Bandits à valeur dans un espace métrique pour l'optimisation et l'approximation

2.2.1 Bandits contre un adversaire

Dans la section précédente, nous avons examiné des jeux où les deux joueurs se soucient de leurs gains. Nous supposons maintenant que le deuxième joueur ne sait pas ce qu'il se passe. Le joueur 2 sera désormais appelé *nature* ou *environnement* : il n'essaie pas de maximiser son gain, mais joue tour après tour sans stratégie (perceptible). Prenons l'exemple d'un homme qui doit choisir chaque matin s'il prend son parapluie ou non avant de sortir, sans connaître la météo de la journée à venir. Chaque jour, l'environnement peut *être dans* (au lieu de *choisir entre*) deux états différents : ensoleillé ou pluvieux. Bien sûr, le but de cet homme est de prendre son parapluie quand il pleut et de le laisser chez lui quand il fait soleil. Il agit en fonction des actions qui maximisent sa récompense. Au contraire, l'environnement ne se soucie pas du tout du résultat du jeu.

Le modèle que nous utilisons est le suivant : à chaque instant t, le joueur choisit une action $x_t \in \mathcal{X}$ qui découle d'une stratégie précédemment choisie σ_t, et il reçoit ensuite une récompense r_t. Pour tout t, σ_t prend en compte l'historique h_{t-1} des précédentes évaluations $x_1, \dots, x_{t-1}$ et des récompenses $r_1, \dots, r_{t-1}$ afin de choisir $x_t = \sigma_t(h_{t-1})$. $\sigma = (\sigma_t)_{t \in \mathbb{N}^*}$ est appelé une *stratégie*. Notons que la stratégie σ peut être déterministe ou stochastique.

Différents contextes ont été étudiés dans le cadre des bandits antagonistes. Le plus connu de tous est celui des *bandits manchots* (Thompson, 1933; Robbins, 1952). Dans les bandits manchots, $\mathcal{X}$ est un ensemble fini avec n éléments, x_t est une action (ou un bras) de $\mathcal{X}$ choisie par le joueur, et la récompense r_t suit une loi μ_{x_t}. L'objectif du joueur est de

trouver l'action $x \in \mathcal{X}$ tel que μ_x ait la plus grande espérance. Des hypothèses doivent être vérifiées par les lois μ_x pour toutes les actions $x \in \mathcal{X}$. Souvent, les lois μ_x sont considérées comme des variables de Bernoulli indépendantes avec un paramètre $p_x \in (0,1)$.[2] Des cas plus généraux examinent des variables indépendantes bornées ou même sous-gaussiennes.[3] Dans le cas plus général des bandits convexes (Flaxman et al., 2005), la récompense r_t prend la forme $g_t(x_t)$ pour certaines fonctions concaves $(g_t)_{1 \leqslant t}$ inconnues du joueur, ayant une valeur uniformément bornée.

Notons qu'à chaque instant t, le joueur ne reçoit comme information que $g_t(x_t)$. Dans le problème plus simple de l'apprentissage avec des experts, le joueur peut avoir une connaissance complète de la fonction de récompense g_t. Quand l'information reçue est complète, les méthodes et algorithmes les plus célèbres sont tous des méthodes du premier ordre : ils utilisent le gradient de la fonction. C'est par exemple le cas de la descente de gradient séquentielle (Online Gradient Descent) de Zinkevich (2003), et sa généralisation, la descente miroir séquentielle (Online Mirror Descent) (Beck and Teboulle, 2003; Shalev-Shwartz and Singer, 2006), ainsi que l'algorithme des poids exponentiels (Littlestone and Warmuth, 1994; Vovk, 1990). Notons que la plupart de ces méthodes ressemblent fortement à celles présentées dans la section 2.1.4. Cela découle du fait que les stratégies pour ces deux types de problèmes sont inspirées par les méthodes de premier ordre de l'optimisation convexe.
Contrairement à l'apprentissage séquentiel, dans le problème des bandits manchots, on ne peut utiliser que des méthodes *d'ordre zéro* : le joueur n'a pas accès aux gradients de la fonction, mais seulement à ses évaluations (parfois stochastique).

Dans le problème d'optimisation convexe de bandits, tout joueur souhaite trouver les actions optimales $x_1^\star, \dots, x_T^\star$ qui maximisent son bénéfice moyen total $\mathbb{E}\left[\sum_{t=1}^T g_t(x_t)\right]$ face à toutes éventuelles situations de l'environnement, c'est-à-dire, face à toutes les séquences adverses $(g_t)_{t \in \mathbb{N}^*}$ de fonctions concaves uniformément bornées auxquelles il pourrait être confronté. Pour simplifier, supposons que g_t a des valeurs dans $[0,1]$ et que sa valeur maximale est 1. Nous notons par $x_t^\star$ un point atteignant la valeur maximal de g_t (avec une valeur égale à 1). Maintenant, le joueur souhaite se comparer au gain optimal *a posteriori*, qui vaut T dans notre exemple. Ainsi, il aurait à déplorer une perte moyenne de gain de $\sum_{t=1}^T \mathbb{E}\left[(g_t(x_t^\star) - g_t(x_t))\right] = T - \sum_{t=1}^T \mathbb{E}\left[g_t(x_t)\right]$ pour la suite de récompense $(g_t)_{1 \leqslant t \leqslant T}$. Maintenant, imaginons que g_t prenne indépendamment de l'historique, et de manière uniformément aléatoire, la valeur $g_t : x \to x$ ou $g_t : x \to 1 - x$. Dans ce cas, le joueur n'a aucun moyen de deviner quelle est la meilleure stratégie σ, et il recevra une rémunération de $1/2$ en moyenne à chaque tour. Cela entraîne une perte de gains de $T/2$, ce qui est du même ordre que T, la pire perte de gains possible. Ceci explique pourquoi, dans la littérature sur les bandits, tous les résultats utilisent une borne sur le regret : la perte de gains par rapport à la meilleure action *fixée* possible.

[2] Une variable X est une variable de Bernoulli si $\mathbb{P}(X = 1) = 1 - \mathbb{P}(X = 0) = p$.
[3] Une variable aléatoire réelle X est dite v-sous-gaussienne si $\mathbb{E}[\exp(\lambda X)] \leqslant \exp(\lambda^2 v/2)$ pour tout $\lambda \in \mathbb{R}$.

Nous notons par R_T^{c} le *regret cumulé* défini par

$$R_T^{\mathrm{c}} = \mathbb{E}\left[\max_{x\in\mathcal{X}} \sum_{t=1}^{T} g_t(x) - \sum_{t=1}^{T} g_t(x_t)\right] .$$

Nous définissons également le *regret simple* : le regret de la $T+1^{\text{ème}}$ évaluation après avoir déjà observé T récompenses du jeu en T points d'évaluations choisis par le joueur.

$$R_T^{\mathrm{s}} = \mathbb{E}\left[\max_{x\in\mathcal{X}} g_{T+1}(x) - g_{T+1}(x_{T+1})\right] .$$

Pour une rémunération constante g (c'est-à-dire $g_t = g$ pour tout $t \leqslant T+1$), le regret cumulé équivaut à la somme de tous les regrets pour les temps $1 \leqslant t \leqslant T$. Pour minimiser cette stratégie, un agent devrait donc toujours faire attention à l'action qu'il choisit à tout instant t et veiller à ne pas avoir un regret important par rapport à la meilleure action fixe. Au contraire, pour le regret simple, seule la dernière action compte. Ainsi, le joueur peut librement explorer des régions avec peu de récompense pour obtenir plus d'informations, sans subir aucune pénalité. Pour cette raison, le regret cumulé est une notion plus forte que le regret simple à un facteur $1/T$ de remise à l'échelle près. Cela est formellement démontré par l'inégalité suivante : $\mathbb{E}[R_T^{\mathrm{s}}] \leqslant \frac{\mathbb{E}[R_T^{\mathrm{c}}]}{T}$.

Maintenant, regardons une stratégie d'exploration-exploitation qui offre de bonnes performances en termes de regret cumulé. L'étude de cet algorithme nous aidera à mieux comprendre les motivations derrière les algorithmes que nous aborderons plus tard dans des contextes de bandits plus complexes.

L'algorithme de la borne supérieure de confiance : Nous présentons maintenant et expliquons intuitivement le fonctionnement de l'algorithme de la borne supérieure de confiance (UCB, de Upper Confidence Bound) étudié par Auer et al. (2002) pour le problème des bandits manchots. Pour des raisons de simplicité, nous considérons un problème de bandits manchots où chaque action $x \in \mathcal{X}$ est liée à une variable aléatoire inconnue μ_x avec support dans $[0,1]$. Nous notons $f(x)$ son espérance (définie pour tout x). Avec cette configuration, f est une fonction de l'ensemble fini $\mathcal{X}$ vers l'ensemble borné $[0,1]$, mais elle sera ensuite étendue à des ensembles d'entrées infinis. Pour résoudre le problème des bandits manchots, une première possibilité serait d'utiliser un algorithme de type Fictitious Play qui choisit à chaque instant t l'action x ayant la meilleure moyenne empirique $\widehat{f}^t(x)$. Cela signifie qu'à tout instant t, $x_t = \arg\max_{x\in\mathcal{X}} \widehat{f}^t(x)$ avec $\widehat{f}^t(x) = \frac{\sum_{s=1}^{t} \mathbb{1}_{\{x_s=x\}} g_s(x_s)}{T_{x,t}}$, où $T_{x,t} = \sum_{s=1}^{t} \mathbb{1}_{\{x_s=x\}}$ et $g_s(x_s) \sim \mu_{x_s}$. Cet algorithme naïf, appelé Follow-The-Leader, exploite pleinement les données, mais n'explore aucunement là où il considère qu'aucun gain est possible. Ainsi, lorsqu'il se trompe, il ne demande pas plus d'informations, mais reste avec sa mauvaise perception de l'environnement et continue de recevoir une récompense sous-optimale, faisant augmenter son regret.

Pour ajouter de l'exploration, une idée serait de prendre $x_t = \arg\max_{x\in\mathcal{X}} u_{x,t}$ où $u_{x,t}$ est de la forme $u_{x,t} = \widehat{f}^t(x) + R(T_{x,t})$ avec R une fonction décroissante. R joue le rôle d'une fonction de régularisation. La question est maintenant de trouver la bonne fonction

R. En effet, si elle décroît trop lentement, nous serons toujours dans le cas précédent sans exploration, mais si elle décroît trop rapidement, l'algorithme n'exploitera jamais et choisira des actions proches d'une distribution uniforme.

Le choix de ce terme supplémentaire est résolu par l'inégalité de Hoeffding, une inégalité de concentration pour les variables bornées.[4] En effet, elle nous dit que pour toute action $x \in \mathcal{X}$,

$$\mathbb{P}\left(f(x) - \widehat{f}^t(x) \geqslant \varepsilon\right) \leqslant \exp(-2T_{x,t}\varepsilon^2)\,,$$

donc pour $\delta = \exp(-2T_{x,t}\varepsilon^2)$,

$$\mathbb{P}\left(f(x) - \widehat{f}^t(x) \leqslant \sqrt{\frac{\ln(1/\delta)}{2T_{x,t}}}\right) \geqslant 1-\delta\,.$$

Nous pouvons obtenir le même type d'inégalité pour $\widehat{f}^t(x) - f(x)$ au lieu de $f(x) - \widehat{f}^t(x)$. Cela nous permet de majorer avec grande probabilité la moyenne de chaque action en fonction de la moyenne empirique, où la borne dépend du nombre $T_{x,t}$ de fois où l'action x a été choisie avant le temps t. On obtient alors UCB (Algorithme 7).

Algorithm 7 Upper Confidence Bound

Entrée: Un ensemble d'entré $\mathcal{X}$.
Initialization: Poser $T_{x,0} = 0$ pour tout x et $u_{x,0} = +\infty$.

pour $t = 0, \ldots,$ **faire**
  Choisir $x_t = \arg\max_{x\in\mathcal{X}} u_{x,t-1}$.
  Observer r_t.
  Poser $T_{x_t,t} = T_{x_t,t-1} + 1$, $\widehat{f}^{t-1}(x_t) = \frac{T\widehat{f}^{t-1}(x_t)+r_t}{T+1}$ et $u_{x_t,t} = \widehat{f}^t(x_t) + \sqrt{\frac{\ln(1/\delta)}{2T_{x,t}}}$.
  Pour toutes les autres actions x, poser $T_{x,t} = T_{x,t-1}$, $\widehat{f}^t(x) = \widehat{f}^{t-1}(x)$ and $u_{x,t} = u_{x,t-1}$.
fin pour

Cet algorithme est un exemple d'*optimisme* dans le contexte des bandits, car il maximise $u_{x,t}$ pour chaque action, qui se trouve être une borne supérieure de la moyenne. L'optimisme[5] est souvent utilisé dans les diverses branches des problème de bandits, comme nous le verrons plus tard.

Maintenant, présentons des bornes sur le regret cumulatif de cet algorithme. Elles peuvent se présenter sous deux formes. Le premier type de borne est une *borne dans le*

[4] L'inégalité de Hoeffding pour des variables indépendantes $X_1, \ldots, X_T$ prenant valeur entre a et b affirme que

$$\mathbb{P}\left(\sum_{t=1}^{T} X_t - T\mathbb{E}[X_1] \geqslant \varepsilon\right) \leqslant \exp\left(\frac{2\varepsilon^2}{T(b-a)^2}\right)\,.$$

[5] L'optimisme dans le contexte des bandits n'a pas la même origine ni la même signification que l'optimisme en théorie des jeux (comme, par exemple, l'optimisme dans l'algorithme de montée-descente de gradient optimiste), et ne devrait pas être confondu.

pire des cas par rapport au nombre n d'actions et au nombre T de rounds. Cette borne nous dit que le regret cumulatif de UCB vérifie

$$R_T = O\left(\sqrt{nT\log(T)}\right) .$$

Cela est vrai quelle que soit la distribution des récompenses pour chaque action.
Il existe également des *bornes dépendantes de la distribution.* Soit $\Delta_x = \max_{x' \in \mathcal{X}} f(x') - f(x)$ l'écart entre la récompense moyenne maximale et la récompense moyenne de l'action x. Alors le regret de UCB satisfait

$$R_T \leqslant 2n + \sum_{x:\Delta_x > 0} \frac{8\log(T)}{\Delta_x} .$$

Cette borne peut être bien plus petite que la précédente pour certaines distributions $(\mu_x)_{x\in\mathcal{X}}$ spécifiques, dès que Δ_x est grand pour tout x. C'est le principal avantage des bornes dépendantes de la distribution. La plupart du temps, les bornes qui dépendent de la distribution sont plus difficiles à obtenir, mais elles donnent toujours une meilleure compréhension du comportement de tout algorithme en fournissant des bornes spécifiques pour toutes les distributions différentes, contrairement aux bornes dans le pire des cas qui traitent toutes les distributions différentes comme une seule.

La différence entre les bornes dans le pire des cas et les bornes dépendantes de la distribution existe également dans d'autres contextes. C'est par exemple le cas pour la prédiction avec l'avis d'experts. De Rooij et al. (2014) ont développé un algorithme appelé FlipFlop, qui atteint une borne optimale dans le pire des cas du même ordre que les algorithmes sans paramètre comme AdaHedge (Erven et al., 2011) ou CMBS Cesa-Bianchi et al. (2007), tout en fournissant une meilleure garantie lorsque les données sont plus simples (presque aussi bonne que pour Follow-The-Leader) : il s'adapte aux données en question.

Nous allons maintenant introduire les bandits dans les espaces métriques et voir comment ces bornes indépendantes de la distribution s'appliquent dans ce cadre.

2.2.2 Bandits dans un espace métrique

Un espace métrique est un ensemble $\mathcal{X}$ équipé d'une distance d, c'est-à-dire une fonction $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ qui vérifie les propriétés suivantes :

- $d(x,x) = 0$ pour tout $x \in \mathcal{X}$ (la distance d'un point à lui-même est nulle) ;
- pour tout $x, y, z \in \mathcal{X}$, $d(x,z) \leqslant d(x,y) + d(y,z)$ (elle respecte l'inégalité triangulaire) ;
- $d(x,y) \geqslant 0$ pour tout $x, y \in \mathcal{X}$ (elle est positive) ;
- $d(x,y) = d(y,x)$ pour tout $x, y \in \mathcal{X}$ (elle est symétrique).

Les espaces métriques les plus courants sont les espaces euclidiens muni de la distance induite par la norme euclidienne.

Soit $(\mathcal{X}, d)$ un espace métrique avec $\mathcal{X}$ compact. Le cadre des bandits dans les espaces métriques (Kleinberg et al., 2008, 2019) est le suivant. Soit $f : \mathcal{X} \to \mathbb{R}$ une fonction. À chaque instant t, le joueur choisit une action x_t et reçoit une récompense r_t qui suit une distribution μ_{x_t} de moyenne $f(x_t)$. Dans la suite, nous considérons le regret simple $R_T^{\mathrm{s}} = \mathbb{E}\left[\max_{x \in \mathcal{X}} f(x) - r_{T+1}\right]$.

Lorsque la récompense r_t est déterministe et égale à $f(x_t)$, minimiser le regret simple R_T^{s} après un certain nombre de tours T revient à évaluer la fonction f en T points différents avant de choisir un point x_{T+1} où le joueur pense que f est maximisée. En effet, dans cette configuration, le regret simple vaut $R_T^{\mathrm{s}} = \max_{x \in \mathcal{X}} f(x) - f(x_{T+1})$, et le minimiser revient à maximiser $f(x_{T+1})$. Ainsi, les bandits dans les espaces métriques rejoignent le cadre de l'optimisation globale (ou boîte noire) et l'optimisation bayésienne.

Dans le cas fini (c'est-à-dire lorsque $\mathcal{X}$ est un ensemble fini), il n'y a pas besoin d'hypothèse sur la fonction f. Cependant, dès que le nombre d'actions est infini, avoir des hypothèses devient nécessaire. Dans le cas contraire, trouver le maximum de n'importe quelle fonction est impossible et le regret simple peut être rendu aussi grand que voulu. Ce que nous aimerions savoir, c'est si $f(x')$ est proche de $f(x)$ pour n'importe quel x' dans le voisinage de x. Pour cela, nous supposons qu'il existe une constante $L > 0$ telle que pour tout $x, x' \in \mathcal{X}$, $|f(x) - f(x')| \leqslant L \cdot d(x, x')$. Cette condition est appelée continuité lipschitzienne d'une fonction. Une fonction f qui vérifie cette condition pour un certain L est dite L-Lipschitz.

Ce cadre peut être généralisé de plusieurs manières.
Tout d'abord, la récompense r_t peut être stochastique. Plusieurs hypothèses existent, mais les plus utilisées sont que la distribution μ_x de la récompense a un support borné (avec une moyenne $f(x)$) ou est uniformément sous-gaussienne pour tout $x \in \mathcal{X}$.
Une autre généralisation consiste à remplacer la maximisation dans le regret simple par toute autre fonctionnelle $J : (\mathcal{X} \to \mathbb{R}) \to \mathbb{R}$ (voir par exemple l'introduction de Kiefer (1957)). L'objectif du joueur serait alors d'approcher $J(f)$ pour certaines fonctions J connues en évaluant d'abord f à T valeurs différentes, puis en donnant en sortie la meilleure estimation J_T de $J(f)$ possible. Ce cadre correspond à celui de l'analyse numérique (Stoer et al., 1980). Le regret simple devient alors $|J(f) - J_T|$. Outre le maximum de f, on peut donner comme exemples de telles fonctionnelles la médiane d'une fonction ou bien son intégrale. Nous reviendrons sur ce dernier problème plus tard. Commençons d'abord par nous concentrer sur le regret simple le plus habituel, qu'est la maximisation.

2.2.3 Méthodes pour des bandits à valeur dans un espace métrique

À présent, nous supposons que le joueur souhaite maximiser une fonction $f : \mathcal{X} \to \mathbb{R}$ inconnue et L-Lipschitz définie sur un ensemble compact $\mathcal{X}$. Ce cadre peut également être appelé *simple-fidélité*, par opposition au paramétrage *multi-fidélité* où le joueur peut avoir accès à des α-fidélités de f pour diverses valeurs de α, c'est-à-dire des approximations de f avec un biais maximal α connu. Le paramétrage multi-fidélité sera expliqué plus en détail ultérieurement. Pour la simple-fidélité, la fonction inconnue f peut être évaluée

parfaitement. Ici, $\mathcal{X}$ est un sous-ensemble compact de $\mathbb{R}^n$ pour un certain $n > 0$ fini. Le joueur a également connaissance du nombre réel L (qui est une borne supérieure, mais pas nécessairement égale, au plus petit nombre réel ℓ tel que f soit ℓ-Lipschitz).

Ce paramétrage simple-fidélité possède une riche littérature dans les cadres déterministes ou bayésiens. Nous fournissons ci-dessous un sous-ensemble (non exhaustif) de références connexes. Les algorithmes d'optimisation *sans certificats* ont une très longue histoire, en optimisation convexe (Nesterov, 2004; Boyd and Vandenberghe, 2004; Bubeck, 2015), en optimisation non convexe (Hansen et al., 1992a,b; Jain and Kar, 2017), en optimisation bayésienne (Garnett, 2023), en optimisation stochastique (Spall, 2003; Bonnans, 2019), ou en optimisation de bandit (Munos et al., 2014; Slivkins et al., 2019). Parmi les techniques algorithmiques les plus proches de cet article, nous pouvons mentionner l'algorithme de Piyavskii-Shubert (Piyavskii, 1972; Shubert, 1972) pour l'optimisation Lipschitz d'ordre zéro, ainsi que des variantes discrétisées telles que l'algorithme de séparation et évaluation de Perevozchikov (1990) ou l'algorithme DIRECT de Jones et al. (1993), pour n'en nommer que quelques-uns. Plus récemment, plusieurs variantes ont également été développées dans la communauté des bandits. Comme exemples d'algorithmes de bandits pour des évaluations parfaites ou bruitées (stochastiques) de f nous pouvons citer DOO (Munos, 2011), HOO (Bubeck et al., 2011a), (Sto)SOO (Munos, 2011; Valko et al., 2013), POO (Grill et al., 2015), qui sont tous basés sur une partition hiérarchique du domaine d'entrée, tout comme l'algorithme c.MF-DOO, présenté dans le chapitre 4. Une autre approche de discrétisation (bien que plus complexe du point de vue computationnel) est l'algorithme Zooming pour les espaces métriques (Kleinberg et al., 2008, 2019). Nous renvoyons le lecteur à Munos et al. (2014); Slivkins et al. (2019) pour plus de détails et de références sur les algorithmes de bandit.

Le domaine de l'optimisation bayésienne, en raison de ses riches contributions au paramétrage multi-fidélité, se trouve également proche de notre travail. Pour l'optimisation simple-fidélité, nous pouvons mentionner les travaux fondateurs de Kushner (1964), l'utilisation de la fonction moyenne d'amélioration introduite dans Močkus (1975), et l'algorithme EGO de Jones et al. (1998), ainsi que des preuves sur les taux de convergence dans Bull (2011). La communauté du krigeage a également abordé l'optimisation globale avec des observations bruitées (Forrester et al., 2006; Picheny et al., 2013). Dans Srinivas et al. (2010, 2012), les auteurs conçoivent et étudient un algorithme de bandit basé sur le processus gaussien (GP-UCB) et dérivent des bornes de regret tant sous des hypothèses bayésiennes que déterministes sur la fonction sous-jacente f. Cet algorithme a été adapté ultérieurement dans Contal et al. (2013) pour le cas d'évaluations mini-batch séquentielles en calcul parallèle. Une revue détaillée de la littérature sur l'optimisation bayésienne peut être retrouvée dans Garnett (2023).

Pour donner plus de contexte, nous souhaitons présenter en détail deux algorithmes parmi ceux précédemment mentionnés. Le premier s'appelle Hierarchical Optimistic Optimization (HOO) et a été introduit par Bubeck et al. (2011a). Il s'agit d'un algorithme de séparation et évaluation (une technique classique en optimisation globale) basé sur l'algorithme UCB décrit dans la section 2.2.1.
Avant de donner l'idée du fonctionnement de HOO, nous devons d'abord introduire la

structure basée sur les arbres utilisée dans un tel algorithme de séparation et évaluation. Une telle structure $(X_{h,i})_{h\in\mathbb{N}^*,i\in\{0,\dots,K^h-1\}}$ est basée sur un arbre K-aire $\mathcal{T}=(h,i)_{h\in\mathbb{N},i\in\{0,\dots,K^h-1\}}$ (un arbre où tous les nœuds ont K enfants) pour lequel chaque nœud est relié à une région de l'ensemble complet $\mathcal{X}$. Chaque nœud est associé à une région $X_{h,i}$ qui vérifie que $X_{0,0}=\mathcal{X}$ et que pour tout (h,i), les régions $(X_{h+1,j})_{(h+1,j)\in\mathcal{C}(h,i)}$ forment une partition de $X_{h,i}$, où $\mathcal{C}(h,i)$ est l'ensemble des enfants de (h,i) dans $\mathcal{T}$. Une région $X_{h,i}$ de l'arbre est également appelée une *cellule*. À chaque cellule $X_{h,i}$ est associé un *représentant* $x_{h,i}$ de la cellule : un élément de la cellule qui la représente. Il peut s'agir d'un point arbitraire dans la cellule, mais il peut aussi être choisi de manière plus astucieuse comme le barycentre de la cellule.

Maintenant, présentons l'idée derrière HOO : à chaque étape, HOO parcourt l'arbre de la racine à l'une des feuilles en choisissant toujours les enfants avec la plus grande borne de confiance supérieure, selon une formule fortement inspirée de l'algorithme UCB. Il évalue ensuite la fonction f en un point de la région correspondante de l'espace $\mathcal{X}$ et met à jour la borne supérieure de tous les ancêtres de cette feuille. Pour présenter une borne supérieure sur le regret de HOO, nous devons d'abord introduire certaines définitions.

Definition 2.1 (Points optimaux d'une fonction)**.** Pour tout $\varepsilon\geqslant 0$, nous désignons l'ensemble des points ε-optimaux de f par $\mathcal{X}_\varepsilon := \{x\in\mathcal{X} : f(x_\star)-f(x)\leqslant\varepsilon\}$, où $f(x_\star)$ est n'importe quel maximum de la fonction f.

Definition 2.2 (Nombre de recouvrement)**.** Pour tout $\varepsilon>0$, le nombre d'empaquetage avec rayon ε de $\mathcal{N}(\mathcal{X}',\varepsilon)$ d'un sous-ensemble $\mathcal{X}'\subset\mathcal{X}$ est le plus grand nombre k de points ε-séparés $x'_1,\dots,x'_k\in\mathcal{X}'$, c'est-à-dire tels que $\left\|x'_i-x'_j\right\|>\varepsilon$ pour tout $i\neq j\leqslant k$.[6] Voir la Figure 2.2 pour un exemple visuel.

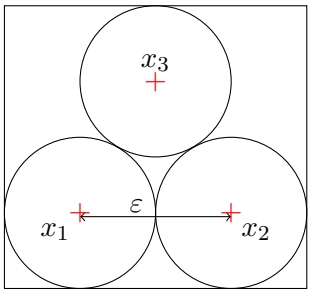


Figure 2.2: Exemple de nombre de recouvrement ε

Definition 2.3 (Dimension de quasi-optimalité)**.** La dimension de quasi-optimalité de f est le plus petit entier n_f tel qu'il existe $C_f>0$ vérifiant : pour tout $\varepsilon>0$, le nombre d'empaquetage $\mathcal{N}(\mathcal{X}_\varepsilon,\varepsilon)$ est inférieur à $C_f\varepsilon^{n_f}$.

En résumé, le nombre d'empaquetage r d'un ensemble $\mathcal{X}'$ est le nombre maximal d'éléments à distance au plus r que l'on peut placer dans l'ensemble $\mathcal{X}'$, et la dimension

[6] Par convention, $\mathcal{N}(\mathcal{X}',\varepsilon)=0$ si $\mathcal{X}'$ est vide. Remarquez également que $\mathcal{N}(\mathcal{X}',\varepsilon)<+\infty$ car $\mathcal{X}$ est compact.

de quasi-optimalité donne un ordre du taux auquel l'empaquetage de la région $\mathcal{X}_\varepsilon$ diminue avec ε. La dimension de quasi-optimalité de n'importe quelle fonction dans $\mathcal{X}$ est inférieure ou égale à n.

Avec ces nouvelles définitions, nous pouvons présenter la borne dans le pire des cas pour HOO donnée dans Bubeck et al. (2011a).

Theorem 2.4. *Soit n_f la dimension de quasi-optimalité de f. Alors, pour tout $n' > n_f$, le regret cumulatif de HOO après T étapes est borné par*

$$\mathbb{E}[R_T^c] \leqslant \gamma T^{(n'+1)/(n'+2)} (\ln T)^{1/(n'+2)}$$

pour une certaine constante γ.

Penchons-nous maintenant sur un autre algorithme, Deterministic Optimistic Optimization (DOO) introduit par Munos (2011). DOO est également un algorithme de type séparation et évaluation basé sur de l'optimisme. Une fois de plus, DOO dépend d'une structure basée sur un arbre $(X_{h,i})_{h\in\mathbb{N}^*, i\in\{0,\dots,K^h-1\}}$ décrite pour HOO. Supposons que le diamètre de chaque région $\mathcal{X}_{h,i}$ est borné par $R\delta^h$ pour un nombre réel R et un $\delta \in (0,1)$. Avec cela en tête, nous pouvons maintenant présenter DOO dans l'Algorithme 8.

Algorithm 8 Deterministic Optimistic Optimization

Entrées: K, $(X_{h,i})_{h\in\mathbb{N}, i\in\{0,\dots,K^h-1\}}$, $(x_{h,i})_{h\in\mathbb{N}, i\in\{1,\dots,K^h-1\}}$, δ, R, L
Initialisation: Soit $t \leftarrow 1$ et $\mathcal{L}_1 \leftarrow \{(0,0)\}$
Choisir le premier nœud $(h^*, i^*) \leftarrow (0,0)$
pour itération$= 1, 2, \dots, T$ **faire**
  pour tout enfant (h^*+1, j) de (h^*, i^*) **faire**
    Fixer $t \leftarrow t+1$ et $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{(h^*+1, j)\}$
    Choisir le point d'évaluation $x_t \leftarrow x_{h^*+1,j}$ et observer $f(x_t)$
  fin pour
  Retirer (h^*, i^*) de $\mathcal{L}_t$
  Fixer $(h^*, i^*) \in \arg\max_{(h,i)\in\mathcal{L}_t} f(x_{h,i}) + LR\delta^h$
fin pour
Renvoyer la recommandation $x_T^* \in \arg\max_{1\leqslant t\leqslant T} f(x_t)$

À chaque instant t, DOO maintient une file $\mathcal{L}_t$ qui contient toutes les feuilles actuelles de l'arbre à explorer. Il choisit parmi les éléments (h,i) de $\mathcal{L}_t$ celui qui maximise $f(x_{h,i}) + LR\delta^h$. Nous le notons (h^*, i^*). Remarquons que pour tout (h,i) et tout $x \in \mathcal{X}_{h,i}$, $f(x)$ est plus petit que $f(x_{h,i}) + LR\delta^h$, car f est L-Lipschitz et le diamètre de $\mathcal{X}_{h,i}$ est majoré par $R\delta^h$. Ainsi, la quantité $f(x_{h,i}) + LR\delta^h$ est une borne supérieure de f sur la région $\mathcal{X}_{h,i}$. Le choix de la région avec la borne supérieure la plus élevée fait de DOO un algorithme optimiste de la même famille que UCB (pour les bandits à bras multiples) et HOO (pour le même problème que DOO). Il développe ensuite la feuille (h^*, i^*), évalue f au point représentatif $x_{h^*+1,j}$ des régions de chacun des enfants de (h^*, i^*), et met à jour $\mathcal{L}_t$ en conséquence.

Un exemple de DOO sur $[0,1]^2$ après quatre époques peut être observé sur la Figure 2.3. Chacune des régions représentées fait partie de $\mathcal{L}_t$, et les croix sont les représentants (le centre de chaque carré a été choisi dans notre exemple, mais n'importe quel représentant fonctionne). L'algorithme devrait se concentrer davantage sur les régions où f a une valeur plus élevée, ce qui laisse penser que le maximum devrait se trouver en haut à droite du centre de $\mathcal{X}$ (proche des cellules actuellement les plus divisées). Cependant, si ce n'était pas le cas, la partie exploration de DOO forcera l'évaluation de régions optimales (ou au moins presque optimales) pas encore visitées de $\mathcal{X}$. Tel quel, DOO ne donne aucune information sur la proximité potentielle d'un point par rapport au maximum, mais nous résoudrons ce problème plus tard avec l'introduction de certificats.

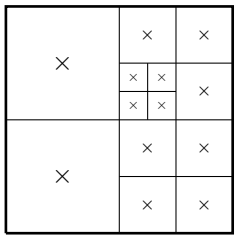


Figure 2.3: Exemple de l'état de DOO après quelques époques.

Passons maintenant à l'énoncé de la borne supérieure du regret simple que l'on peut obtenir en utilisant DOO, comme le montre Munos (2011).[7]

Theorem 2.5. *Si la dimension de quasi-optimalité n_f de f est strictement supérieure à zéro, alors le regret simple diminue de manière polynomiale : $R_T^s \leqslant (LR)^{n_f+1/n_f}(1-\delta^{n_f})^{-1/n_f} C_f^{1/n_f} T^{-1/n_f}$.*
Si la dimension de quasi-optimalité de f égale zéro, alors le regret simple diminue de manière exponentielle : $R_T^s \leqslant LR\delta^{T/C_f} - 1$.

Nous décrivons maintenant trois extensions de cette configuration. La première est la multi-fidélité : le joueur peut évaluer différentes approximations de la fonction f. La deuxième est la certification : comment certifier la sortie d'un algorithme. La dernière est le fait de donner des bornes dépendantes de f. Dans ce cas de figure, les bornes données ne sont plus pire cas, mais elles dépendent de chacune des fonctions f étudiées. Une configuration qui mélange ces trois paramètres sera présentée dans le chapitre 4.

Multi-fidelité

Dans le cadre de l'optimisation bandit dans des espaces métriques avec simple-fidélité, nous avons considéré l'optimisation d'une fonction $f : \mathcal{X} \to \mathbb{R}$ qui pouvait être évaluée de manière parfaite. Nous supposons maintenant que, en plus de la fonction f qu'il cherche à

[7] En fait, le théorème énoncé est plus général, mais nous ne donnons ici qu'un corollaire, pour faciliter la comparaison avec les bornes de HOO.

maximiser, le joueur peut accéder à des approximations $\{f_\alpha|\alpha \in \Lambda\}$ de f.[8] Nous appelons α-fidélité de f une fonction f_α telle que pour tout $x \in \mathcal{X}$, $|f_\alpha(x) - f(x)| \leqslant \alpha$. De plus, nous supposons également l'existence d'une fonction de coût $c : \Lambda \to \mathbb{R}$ telle qu'une évaluation de f_α en tout point $x \in \mathcal{X}$ coûte $c(\alpha)$. Bien sûr, plus la précision α est faible, plus le coût $c(\alpha)$ est élevé : on suppose que c est décroissante. L'objectif du joueur est de trouver un bon compromis entre une bonne précision lors de l'évaluation de f et un faible coût encouru.

Le premier scénario étudié concernait deux fonctions de fidélité : une fonction ayant une grande fidélité f que le joueur souhaite minimiser, et une fonction de moindre fidélité f_α pour un certain α connu, qui est utilisée comme une approximation moins coûteuse de f. À notre connaissance, les premiers articles sur le sujet proviennent de la communauté du krigeage, avec l'étude du cokrigeage (Ver Hoef and Cressie, 1993). Notons cependant que le cokrigeage ne se concentre pas explicitement sur l'optimisation. Quelques années plus tard, plusieurs articles ont utilisé la fonction de faible fidélité pour explorer des zones inconnues, et la fonction de haute fidélité pour exploiter les connaissances antérieures dans des zones avec des valeurs approximatives proches du maximum. Parmi de nombreux exemples, citons-en deux intéressants. Le premier (Eby et al., 1998) utilise un algorithme génétique pour lequel ils se servent de la fonction de moindre fidélité pour peupler la nouvelle génération de l'algorithme. Un deuxième exemple est celui de Alexandrov et al. (1998) pour un algorithme de région de confiance. Au lieu d'approcher la bonne fidélité par une approximation quadratique locale sur la région de confiance, ils ont recours à une fonction de faible fidélité, donnant de bons résultats dès que celle-ci est plus proche de la haute fidélité que son approximation du second ordre.

Ces deux premiers exemples ont ensuite été suivi par de nombreux nouvelles idées d'algorithmes, souvent avec de meilleures bases théoriques et des résultats empiriques plus solides.
De nombreux travaux dans le cadre de la multi-fidélité proviennent de l'optimisation bayésienne. Par exemple, Huang et al. (2006) ont conçu une variante multi-fidélité de l'algorithme EGO ; Zhang et al. (2017) ont généralisé l'algorithme de recherche d'entropie prédictive de Hennig and Schuler (2012); Hernández-Lobato et al. (2014) et Takeno et al. (2020) ont généralisé la recherche d'entropie de valeur maximale (Wang and Jegelka, 2017) ; une contrepartie multi-fidélité de GP-UCB (MF-GP-UCB) a été étudiée dans Kandasamy et al. (2016a, 2019), tandis que le cokrigeage a été étudié pour l'optimisation, par exemple, dans Forrester et al. (2007); Qian and Wu (2008). Un cadre dans lequel les fonctions de fidélité (en nombre fini) peuvent être mutuellement dépendantes a été introduit dans Song et al. (2019), avec une garantie de probabilité élevée. Une revue plus complète de l'optimisation bayésienne multi-fidélité peut être trouvée dans Peherstorfer et al. (2018) et dans Garnett (2023, Section 11.5).

Récemment, les algorithmes de bandits ont également été étendus au cadre multi-fidélité. (Kandasamy et al., 2016b) a étudié dans un premier temps un nombre fini de

[8] Λ était supposé fini dans les premiers articles sur le sujet, mais a rapidement été étendu à des ensembles potentiellement l'infini dans Sen et al. (2018).

bras. Plusieurs algorithmes ont ensuite été conçus pour un nombre continu de bras, ce qui correspond à l'optimisation multi-fidélité *sans* certificats. Une extension de DOO (appelée MFDOO) a été proposée par Sen et al. (2018), avec une borne d'erreur d'optimisation (ou de regret simple) pour un budget de coût global donné. Les auteurs ont également étudié une variante inspirée de POO (MFPDOO) pour traiter le cas d'une régularité inconnue (voir le paragraphe ci-dessous). Des algorithmes similaires basés sur HOO et POO (appelés MFHOO et MFPOO) ont été introduits ultérieurement par Sen et al. (2019) pour faire face à un bruit additionnel (stochastique). Dans Fiegel et al. (2020), les auteurs développent l'algorithme Kometo (basé sur StroquOOL) et démontrent des bornes supérieures presque optimales (avec des bornes inférieures minimax correspondantes) sur l'erreur d'optimisation, compte tenu d'un budget de coût global. Leur analyse couvre également les cas de régularité inconnue et de coûts potentiellement non bornés à des précisions α dans le voisinage de $\alpha = 0$. Le cas d'un retour d'information bruité et avec délai a été abordé dans Wang et al. (2022), dans lequel ils utilisent une généralisation de HOO.

Un exemple typique d'un problème réel multi-fidélité est l'optimisation d'une fonction f calculée avec la méthode des éléments finis. Un exemple avec deux fonctions de fidélité (deux valeurs d'α) apparaît dans Sun et al. (2011), pour la conception de tôle avec pour objectif de ne pas avoir de défauts dans les produits (le cas d'un panneau intérieur de voiture dans cet article). Étant donné trois variables x_1, x_2, x_3 modélisant des forces de retenue importantes sur le métal, l'objectif est de régler ces forces à une bonne valeur pour éviter à la fois la rupture et le plissement. Deux solveurs d'éléments finis différents ont été utilisés pour approcher f en tout point x : des solveurs d'éléments finis incrémentaux, et un modèle d'éléments finis en une étape, qui est computationnellement moins coûteux mais fournit de moins bonnes estimations que le premier modèle. Trouver une bonne conception des forces à un coût de calcul raisonnable est un exemple de problème d'optimisation multi-fidélité (à deux niveaux). De nombreux autres exemples peuvent être trouvés, par exemple, en thermodynamique Dewettinck et al. (1999); Le Gratiet (2013), dans la conception de nouveaux avions Geiselhart et al. (2011), ou dans la sécurité du nucléaire Picheny et al. (2010). Une revue avec de nombreuses applications de multi-fidélité peut être vue dans Fernández-Godino et al. (2016).

Dans le chapitre 4, nous étudions un cadre plus large avec un nombre potentiellement infini de fonctions de fidélité. Plus formellement, nous prenons $\Lambda = \mathbb{R}^*$ et $c : \mathbb{R}^* \to \mathbb{R}$ une fonction croissante. Remarquons que dans cet exemple, on ne peut pas évaluer la fonction originale f car 0 n'est pas dans Λ. En fait, la plupart de nos résultats peuvent être étendus au cas $\Lambda = \mathbb{R}$ dès que c possède une limite à droite en 0, mais pour simplifier la rédaction et une meilleure lisibilité, nous ne couvrirons pas ce cas. Chaque fois que le coût c est une fonction en escalier avec un nombre fini de morceaux, on revient à la configuration précédente avec un nombre fini de fidélités (auquel cas on peut évaluer f avec un coût fini).

De la nécessité d'un certificat

En pratique, les algorithmes qui atteignent de petites erreurs d'optimisation sont souhaitables, mais peuvent ne pas informer l'utilisateur lorsque de telles erreurs d'optimisation ont été obtenues. Dans l'exemple ci-dessus, un ingénieur pourrait avoir besoin de *certifier* ce que renvoie l'algorithme, c'est-à-dire obtenir une borne de garantie sur l'erreur d'optimisation qu'il peut calculer en utilisant uniquement les données observées et certaines hypothèses connues sur f, comme cela est fait, par exemple, dans Hansen et al. (1991) et Bachoc et al. (2021) dans le cadre de la simple-fidélité. Une telle exigence peut être importante dans les domaines industriels impliquant des systèmes critiques pour la sécurité (par exemple, les voitures, les avions, la santé, le génie nucléaire).

Bien que des résultats de convergence concernant l'erreur d'optimisation $f(x_\star) - f(x_t^*)$ en fonction d'un budget de coût total soient maintenant bien établis dans le cadre de la multi-fidélité, la question de la *certification* des maximiseurs approximatifs avec un coût total minimum n'a pas été abordée, à notre connaissance.

La notion de certificat d'erreur est apparue dans plusieurs autres contextes, tels que, par exemple, en optimisation convexe (Boyd and Vandenberghe, 2004), où l'*écart de dualité* entre les points faisables primaires et duaux joue le rôle d'un certificat d'erreur. Dans l'optimisation Lipschitzienne avec évaluations parfaites de f (cadre de la simple-fidélité), l'algorithme de Piyavskii-Shubert (Piyavskii, 1972; Shubert, 1972) est naturellement doté d'un certificat d'erreur, qui est la différence entre la valeur maximale d'une fonction de majoration supérieure garantie de f et la valeur maximale $f(x_s)$ observée jusqu'à présent. Pour les entrées unidimensionnelles, une analyse précise du nombre d'évaluations avant que ce certificat ne tombe en dessous de ε a été donnée dans Hansen et al. (1991) (voir également Danilin (1971)), avec une simple expression intégrale. Ce résultat a été généralisé aux entrées multidimensionnelles par Bouttier et al. (2020); Bachoc et al. (2021), et montré dans Bachoc et al. (2021) comme réalisable avec l'algorithme c.DOO de manière efficace[9], avec une borne f-dépendante inférieure correspondant presque parfaitement avec la borne f-dépendante supérieure. Dans cet article, nous étendons l'analyse de complexité de Bachoc et al. (2021) au cadre de la multi-fidélité, en utilisant une variante certifiée de l'algorithme MFDOO Sen et al. (2018) pour la borne supérieure.

Une autre série importante de travaux concerne l'adaptabilité à la régularité (inconnue) de f, c'est-à-dire savoir réaliser des performances d'optimisation presque optimales avec un algorithme qui n'a (presque) aucune connaissance préalable de la régularité de f. Parmi les nombreux algorithmes conçus dans ce but, mentionnons l'algorithme DIRECT de Jones et al. (1993), l'algorithme $Z(k)$ de Horn (2006), ainsi que les algorithmes bandit (avec des garanties de regret simple ou cumulatif), notamment (Sto)SOO (Munos, 2011; Valko et al., 2013), l'algorithme deux-phases de Bubeck et al. (2011b), POO et GPO (Grill et al., 2015; Shang et al., 2019), AdaLIPO (Malherbe and Vayatis, 2017), SequOOL et StroquOOL (Bartlett et al., 2019), et leurs variantes multi-fidélité (Sen et al., 2018, 2019; Fiegel et al., 2020). Voir aussi Locatelli and Carpentier (2018) pour un compte rendu détaillé des résultats d'adaptabilité possibles et impossibles dans le cadre de la fidélité

[9] C'est-à-dire avec un petit nombre (logarithmique) d'opérations élémentaires par évaluation de f.

unique.

Bien que l'adaptabilité à la régularité soit une caractéristique clé de robustesse des algorithmes d'optimisation, nous soulignons qu'elle est d'une certaine manière *incompatible* avec l'exigence de certification. Par exemple, comme l'a noté Bachoc et al. (2021) dans le cadre de la fidélité unique, lors de l'optimisation d'une fonction f Lipschitzienne avec une constante Lipschitz inconnue $\mathrm{Lip}(f)$, il est impossible de produire un certificat fini ξ_t après un nombre quelconque t d'évaluations de f, car il pourrait y avoir une bosse arbitrairement raide dans une région d'entrée non observée. Plus formellement, si f a un maximiseur $x_\star$ à l'intérieur de $\mathcal{X}$, la borne inférieure de (Bachoc et al., 2021, Théorème 2) croît au moins comme $\left(L/\mathrm{Lip}(f)\right)^n$ lorsque $L \to +\infty$, ce qui implique que le nombre minimum d'évaluations nécessaires pour un algorithme certifié pour une fonction f est arbitrairement grand si nous demandons à un tel algorithme de produire des certificats valides pour toutes les fonctions Lipschitz g ayant des constantes de Lipschitz $\mathrm{Lip}(g)$ arbitrairement grandes. La même remarque intuitive s'applique à notre cadre multi-fidélité, selon la borne inférieure du Théorème 4.3 dans la section 4.3.

2.2.4 Estimation séquentielle d'intégrales

Comme mentionné à la fin de la section 2.2.2, le cadre des bandits dans un espace métrique peut être modifié pour un objectif autre que la maximisation. Son nouvel objectif est alors d'approcher une fonctionnelle $J(f)$ de la fonction f étudiée. Dans cette section, la fonctionnelle que nous considérons est l'intégration. Plus formellement, soit $\mathcal{X}$ un ensemble compact de $\mathbb{R}$ et μ une mesure de probabilité sur $\mathcal{X}$. Pour toute fonction intégrable par rapport à μ, nous définissons $J(f)$ comme étant $\int_{\mathcal{X}} f(x)\,\mathrm{d}\mu(x)$.

Lorsque $\mathcal{X}$ est le segment $[0,1]$, plusieurs méthodes bien connues peuvent être utilisées. Supposons d'abord que l'horizon est fini et connu à l'avance par le joueur. Nous le fixons à $T+1$ pour un certain T dans $\mathbb{N}$. Une première idée consisterait à évaluer f sur une grille uniforme de $[0,1]$. Soit $x_i = \frac{i}{T}$ pour $i = 0, \dots, T$ les $T+1$ points évalué. Ensuite, pour approcher f sur tous les petits sous-intervalles $([x_{i-1}, x_i])_{1 \leqslant i \leqslant T}$, on pourrait utiliser :

- La méthode du point de gauche (respectivement, du point de droite) : elle approche f par $x \mapsto f(x_{i-1})$ (respectivement, $x \mapsto f(x_i)$) sur le segment $[x_{i-1}, x_i]$. Ces deux méthodes donnent une erreur de l'ordre de $1/T$ pour les fonctions de $\mathcal{C}([0,1])$.

- La méthode du point milieu (ou médian) : elle approche f par $f\left(\frac{x_{i-1}+x_i}{2}\right)$, l'évaluation de f au milieu de chaque segment. Elle donne une erreur de l'ordre de $1/T^2$ dès que la fonction est $\mathcal{C}^2$ sur $[0,1]$.

- La méthode des trapèzes : elle approche f par son interpolation linéaire

$$\widehat{f} \colon x \mapsto \frac{x - x_{i-1}}{x_i - x_{i-1}}(f(x_i) - f(x_{i-1})) + f(x_{i-1})\,.$$

  Ensuite, l'intégrale de $\widehat{f}$ sur $[x_{i-1}, x_i]$ vaut $\frac{f(x_{i-1})+f(x_i)}{2}$. Comme pour la méthode du point milieu, on obtient une erreur de l'ordre de $1/T^2$ pour les fonctions $\mathcal{C}^2$. Les constantes (non exprimées ici) sont pires que celles de la méthode du point milieu,

cependant la méthode des trapèzes n'a pas besoin d'une autre évaluation sur chaque sous-intervalle, contrairement à la méthode du point milieu.

- Un dernier algorithme largement utilisé qui fonctionne avec la grille uniforme est la méthode de Simpson, qui donne une erreur de l'ordre de $1/T^4$ pour les fonctions $\mathcal{C}^4$.

Un autre exemple avec une grille non uniforme de points $x_0, \ldots, x_T$ est la méthode de la quadrature de Gauss, une approximation basée sur les polynômes de Legendre. Chaque nœud x_i et poids w_i (associé à $f(x_i)$) peut être calculé avec l'algorithme de Golub-Welsch (Golub and Welsch, 1969). On pourrait également utiliser des méthodes qui n'utilisent pas de combinaison affine des évaluations de f pour approcher son intégrale.

Toutes les méthodes précédentes sont non adaptatives : chaque point n'est pas choisi séquentiellement pour minimiser l'erreur, mais est choisi à l'avance de manière offline (pour un horizon fixe $T+1$). On pourrait également penser à des méthodes adaptatives, qui choisissent séquentiellement chaque point d'évaluation. À une étape donnée $t \leqslant T$, elle peut alors utiliser l'une des méthodes précédemment énoncées pour estimer f sur tous les sous-intervalles $([x_{i-1}, x_i])_{1 \leqslant i \leqslant t}$. La seule différence est que les sous-intervalles peuvent ne pas être uniformes. De telles méthodes incluent les précédentes, élargissant ainsi le champ des possibilités.

Par exemple, imaginons le cadre de jeu suivant où l'on doit estimer l'intégrale d'une fonction constante par morceaux f avec seulement deux morceaux sur $[0,1]$. Notons par $x_\star \in [0,1]$ la seule discontinuité inconnue de f. Ensuite, en utilisant la méthode des trapèzes (grille uniforme avec interpolation affine), la borne dans le pire des cas est de l'ordre de $1/T$. Cependant, en utilisant une méthode séquentielle basé sur de la dichotomie pour trouver $x_\star$ et ensuite une interpolation affine, on obtient une borne dans le pire des cas de l'ordre de $\exp(-T)$, nettement plus rapide que la méthode précédente.

De cet exemple, on pourrait penser que les méthodes adaptatives sont toujours nettement meilleures que les méthodes non adaptatives, car elles offrent plus de flexibilité et peuvent s'adapter à chaque fonction (comme leur nom l'indique). Cependant, il a été prouvé que dans certains cas, elles n'améliorent pas l'erreur d'estimation pire cas ! En effet, notons $\mathcal{F}$ l'ensemble des fonctions étudiées (par exemple, l'ensemble des fonctions intégrables de $\mathcal{X}$ vers $\mathbb{R}$). Alors, Bakhvalov (1971) a montré que si l'ensemble $\mathcal{F}$ est convexe et symétrique, les méthodes adaptatives ne sont pas meilleures que les méthodes non adaptatives. Dans le même sens, Sukharev (1986) a prouvé que pour un ensemble convexe $\mathcal{F}$, les méthodes non adaptatives ne sont pas meilleures que les méthodes affines. C'est pourquoi, dans de nombreux contextes, bien que simples, les méthodes précédemment données fournissent une erreur d'estimation minimale de l'intégrale.

Une autre extension des méthodes précédentes doit être abordée : les méthodes aléatoires. La randomisation peut apparaître à la fois dans le choix des points d'évaluations ou dans l'approximation de f sur chaque intervalle. Ici, nous nous concentrons sur le deuxième type de randomisation. Un tel exemple peut être vu dans la méthode des trapèzes aléatoires (Wu, 2022). Dans cet article, l'auteur montre que la convergence de la méthode des

trapèzes a été améliorée d'un facteur $T^{1/2}$ avec la randomisation pour toute fonction dans les espaces de Sobolev. Cependant, la randomisation n'améliore pas les résultats pour toutes les classes de fonctions. Maintenant, étudions le cas spécifique des fonctions monotones, voyons comment nous pouvons appliquer la littérature précédente à ce contexte, et si nous pouvons fournir de nouvelles bornes.

2.2.5 Approximation et estimation d'intégrales de fonctions monotones

Dans le chapitre 5, nous étudions spécifiquement les *fonctions monotones*. Nous étudions l'estimation d'intégrale de fonctions monotones dans la section 5.4.1, avec l'introduction et l'étude théorique de GreedyBox, un algorithme optimal (à facteurs logarithmiques près) grandement inspiré du travail de Novak (1992). Nous améliorons également en espérance l'erreur d'estimation à $T^{3/2}$ avec une version stochastique de cet algorithme, qui atteint la borne supérieure pire cas des méthodes adaptatives aléatoires prouvées par Novak (1992).

Remarquons que l'ensemble $\mathcal{F}$ des fonctions monotones sur un segment est convexe mais non symétrique. Ainsi, selon les résultats présentés dans la section précédente, nous savons que les méthodes non adaptatives ne sont pas meilleures que les méthodes affines sur cet ensemble. Kiefer (1957) a prouvé que parmi les méthodes déterministes, la méthode des trapèzes est optimale, montrant que même si l'ensemble des fonctions monotones n'est pas symétrique, les résultats de Bakhvalov (1959) s'appliquent ici. Novak (1992) a ensuite démontré que pour l'estimation intégrale, les méthodes adaptatives aléatoires peuvent améliorer à une erreur pire cas de l'ordre de $1/T^{3/2}$ la borne précédente de $1/T$ trouvée pour toutes les méthodes déterministes et les méthodes aléatoires non adaptatives.

Pour aller plus loin, les bornes précédentes ont été ultérieurement étendues dans Papageorgiou (1993) à l'approximation d'intégrale de fonctions monotones à plusieurs variables. Les livres de Davis and Rabinowitz (2007) et de Brass and Petras (2011) offrent un éventail plus large de résultats sur l'intégration numérique pour diverses hypothèses.

Bien sûr, de nombreux autres ensembles de fonctions ont été étudiés. Une liste non exhaustive comprend les travaux sur l'ensemble des fonctions unimodales[10] de Novak and Roschmann (1996), sur l'ensemble des fonctions à variation bornée[11] de Graf and Novak (1990), sur les classes convexes et symétriques de fonctions de (Novak, 1993, 1995; Hinrichs et al., 2011), et sur diverses classes de fonctions à plusieurs variables de (Ritter et al., 1993; Katscher et al., 1996; Krieg and Novak, 2017). Toutes les bornes données dans tous ces articles ne dépendent pas de f.

De l'estimation d'intégrale à l'approximation : Tous les travaux précédents portent sur l'estimation d'intégrale, un problème plus simple que l'approximation L^p. Plusieurs articles étudient l'approximation L^p de diverses classes de fonctions. Un exemple connu

[10] Une fonction f est dite unimodale s'il existe un $x \in [0,1]$ tel que f soit croissante sur $[0,x]$ et décroissante sur $[x,1]$.

[11] Une fonction f est de variation bornée si elle peut être écrite comme la différence de fonctions croissantes

est l'interpolation polynomiale pour les fonctions k fois continûment différentiables. De nombreux travaux utilisent le module de régularité de la fonction f pour encadrer son approximation L^p, fournissant des bornes dépendantes de f. C'est par exemple le cas de Chandra (2002) pour l'approximation de fonctions périodiques à l'aide de polynômes trigonométriques. Le calcul de la meilleure approximation linéaire L^p, où une base $\phi_1, \dots, \phi_k$ de fonctions est donnée au préalable, et où l'on cherche les poids $w \in \mathbb{R}^k$ minimisant $\|f - \sum_i w_i \phi_i\|_p$, a été largement étudié (Fletcher et al., 1971, 1974). Certains travaux portent sur les fonctions k-monotones, c'est-à-dire les fonctions avec une dérivée k-ième monotone. Kopotun and Shadrin (2003); Kopotun et al. (2009) ont étudié le taux de convergence des méthodes d'interpolation sur cet ensemble de fonctions, et ont montré des résultats dépendant du module de régularité de la fonction. À notre connaissance, aucun travail n'a été réalisé dans le cas de l'approximation $L^p(\mu)$ pour des fonctions croissantes dans leurs généralités, sans hypothèse de continuité ou de régularité.

Comme l'a indiqué Novak (1996), l'adaptation dépend de la connaissance *a priori* du problème. L'adaptation en général est au cœur de plusieurs algorithmes de pointe pour les problèmes séquentiels. Un premier exemple est l'utilisation de la méthode de Newton pour le problème de recherche de zéro. Un autre grand domaine pour les méthodes adaptatives est celui de l'apprentissage en ligne et des problèmes de bandits (Cesa-Bianchi and Lugosi, 2006). Des exemples dans la littérature sur les bandits incluent l'algorithme MOSS de Audibert et al. (2009) ou KL-UCB de Garivier and Cappé (2011), ou encore AdaHedge De Rooij et al. (2014) pour le problème d'apprentissage avec conseils d'experts. L'adaptation apparaît également dans l'optimisation globale de fonctions boîte noire, avec des algorithmes tels que celui de Piyavskii-Shubert (Piyavskii, 1972; Shubert, 1972) ou POO (Grill et al., 2015). Un autre travail dans ce domaine est celui de Bachoc et al. (2021), où les auteurs ont donné des bornes dépendantes de f en introduisant la notion de certificat dans le cadre de l'optimisation boîte noire.

Dans les sections 5.2 et 5.3, nous étudions le problème de l'approximation $L^p(\mu)$ pour les fonctions monotones. Étant donné une fonction monotone f, nous ne voulons plus trouver une approximation de $J(f)$, mais une approximation de f elle-même, où l'erreur est calculée avec la norme $L^p(\mu)$ définie par $\|f-g\|_p = \left(\int_{x\in\mathcal{X}} |f(x)-g(x)|^p \,\mathrm{d}\mu(x)\right)^{\frac{1}{p}}$. Lorsque $p=1$, obtenir une bonne approximation de f implique de fournir une bonne estimation de l'intégrale. GreedyBox fournit la même borne que pour l'estimation d'intégrale lorsque $p=1$, et la borne trouvée peut être généralisée à tout $p \geqslant 1$. Les bornes supérieures que nous fournissons sont dépendantes de f et correspondent, jusqu'à des facteurs logarithmiques près, aux bornes inférieures correspondantes.

Chapter 3

Optimistic Gradient Descent Ascent in Zero-Sum and General-Sum Bilinear Games

We study the convergence of Optimistic Gradient Descent Ascent (OGDA) in unconstrained bilinear games. For zero-sum games, we prove the exponential convergence of OGDA to a Nash equilibrium for any payoff matrix, and provide the exact ratio of convergence as a function of the step size. Then we introduce OGDA for general-sum games, and show that in many cases, either OGDA converges exponentially fast to a Nash equilibrium, or the payoffs for both players converge to $+\infty$. We also show how to increase drastically the speed of convergence of a zero-sum problem, by introducing a general-sum game using the Moore-Penrose inverse of the original payoff matrix. To our knowledge, this shows for the first time that general-sum games can be used to optimally improve algorithms designed for min-max problems.

This chapter is based on a joint work with Jérôme Renault.

Contents

3.1 Introduction 53
3.2 Zero-sum bilinear games 56
3.2.1 Convergence of OGDA for zero-sum games 56
3.2.2 Optimal geometric ratio of convergence 57
3.3 General-sum bilinear games 57
3.3.1 Sufficient conditions for convergence in general-sum games 58
3.3.2 Improving the zero-sum rate of convergence with general-sum games 59
3.3.3 "Cooperation" induced by OGDA 60
3.4 Illustration: Generative Adversarial Networks 61
3.4.1 For bilinear games 61
3.4.2 Experimental results for some concave/convex games 63

3.5 Conclusion **64**
3.A Comments on Theorem 3.1 . . . . . **66**
3.B Proof of Theorem 3.1 . . . . . **67**
3.B.1 Convergence to a Nash equilibrium for general $\eta > 0$ 68
3.B.2 Convergence to a Nash equilibrium if $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$ 71
3.B.3 Convergence to a Nash equilibrium if $\frac{1}{4\eta^2} \in \mathcal{S}(A)$ 71
3.B.4 Speed of convergence when $0 < \eta < \frac{1}{\sqrt{3}\mu_{\max}}$ and $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$ . . . 73
3.B.5 Convergence with the linear terms 77
3.B.6 On the optimality of $\lambda(\eta)$ 78
3.C Proof of Theorem 3.3 . . . . . **79**
3.C.1 Description of the eigenvalues of $\Lambda_{A,B}$ 80
3.C.2 Convergence without linear terms 81
3.C.3 Convergence with linear terms 83
3.D Proof of Theorem 3.4 . . . . . **83**
3.E Proof of Theorem 3.5 . . . . . **84**
3.F Proof of Proposition 3.6 . . . . . **86**

3.1 Introduction

Min-max optimization is receiving a lot of attention, due in particular to the popularity of generative adversarial networks (GANs), introduced by Goodfellow et al. (2014), and of adversarial training (Madry et al., 2018). In all such cases, two entities are in competition: a first player aims at maximizing a loss function that another player wants to minimize. One could think that applying optimization algorithms independently for both player will make them converge to a mutual agreement. However, min-max problems are not that easy, in particular due to their inherent competition which brings instability. The whole goal of any min-max algorithm is to find a way to bring the two competitors to an equilibrium that will suit both of them.

An even harder problem is the one of general-sum games. In this setting, no more assumption is done on the sum of the two payoff functions. When both payoffs are opposite from each other, we come back to the min-max (or zero-sum) case. When the payoffs for both player are the same, cooperation is needed, and may simplify the strategy for both player. In their paper, Goodfellow et al. (2014) proposed to use the non-saturating loss to avoid vanishing gradient. Thus, the two players play in practice a general-sum games. Another example of the use of games with non-zero sum is adversarial machine learning (Dasgupta and Collins, 2019).

Formally, consider a payoff function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ defined on a product set $\mathcal{X} \times \mathcal{Y}$. In the associated zero-sum game, simultaneously player 1 chooses $x \in \mathcal{X}$ and player 2 chooses $y \in \mathcal{Y}$, and finally player 2 pays the quantity $g(x, y)$ to player 1. The game-theoretic

approach is to look for a Nash equilibrium (or saddle point) of g, i.e. for a point (x^*, y^*) in $\mathcal{X} \times \mathcal{Y}$ such that:

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, g(x, y_\star) \leqslant g(x_\star, y_\star) \leqslant g(x_\star, y) \,. \tag{3.1}$$

As was seen in Section 1.1.4 of the introduction, the Gradient Descent Ascent (GDA) algorithm does not always converge to a Nash equilibrium, which brings the need for new algorithms. To this goal Popov (1980) introduced OGDA, an optimistic version of GDA presented in Section 1.1.5 (Algorithm 2). Throughout the chapter we consider unconstrained cases where $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$, and focus on Optimistic Gradient Descent Ascent with constant step size $\eta > 0$ and on bilinear payoffs: $g(x, y)$ is of the form $x^T A y + b^T x + c^T y + d$. Here, the difficulty is not to find a saddle point of g (this is fairly easy if it exists), but to prove or disprove the convergence of OGDA to a saddle point.

Contributions of this paper.

- In Section 3.2, we improve the previous results of convergence for zero-sum bilinear games. Whenever a Nash equilibrium exists, Theorem 3.1 shows that OGDA converges exponentially fast to a Nash equilibrium that we characterize. We also characterize the range of step sizes η where convergence happens and provide a precise speed of convergence. Moreover, we exhibit the *exact geometric ratio* $\lambda(\eta)$ for the convergence, improving the previously obtained ratios. We believe these results of last-iterate convergence matter because they show a strong stability of the OGDA algorithm: not only OGDA can be used to find ε-equilibria, but it will not exhibit cyclic behaviors within the possibly large set of Nash equilibria. Theorem 3.1 can be used to compute the step size $\eta^\star$ optimizing the geometric ratio $\lambda(\eta)$.

- We then consider OGDA for general-sum bilinear games, where the payoff function of player 1 can be written $g_1(x, y) = x^T A y + b^T x + c^T y + d$ and the payoff function of player 2 $g_2(x, y) = x^T B y + e^T x + f^T y + g$. We introduce for the first time the natural extension of OGDA for these general-sum games. We show that OGDA needs not always converge to a Nash equilibrium here, but we give in Theorem 3.3 sufficient conditions for convergence to a Nash equilibrium, when all eigenvalues of $B^T A$ are real negative.

- We then show in Theorem 3.4 one major interest of introducing OGDA for general-sum games: the speed of convergence to a solution of a min-max problem can be optimally improved if one considers a well suited general-sum game, instead of the natural associated zero-sum game. The idea is to introduce a general-sum game where $B = -(A^\dagger)^T$, with $A^\dagger$ the Moore-Penrose inverse of A.

- The last part of Section 3.3 considers OGDA for a particular class of payoff matrices, including in particular the case of common payoff games where $A = B$ (implying that there is a potential for coordination or cooperation among players), or more generally the cases where $B = \alpha A$ for some $\alpha \neq 0$. It is proved that for these games, either OGDA converges to a Nash equilibrium, or under OGDA the payoff for both players tends to $+\infty$ exponentially fast as $t \to +\infty$. This result is a significant extension of the convergence of OGDA, since both players having $+\infty$ payoffs can be seen as a generalized Nash equilibrium of the game and a very desirable outcome of the interaction.

- Finally, in Section 3.4 we illustrate our results on stylized versions of Generative Adversarial Networks, in the spirit of Daskalakis et al. (2018). We show in particular on an example how introducing a general-sum game using the Moore-Penrose inverse $A^\dagger$ improves the speed of convergence.

Other related works. The optimistic version of the Gradient Descent algorithm was first seen in a paper by Popov (1980). Rakhlin and Sridharan (2013) introduced an algorithm called Optimistic Mirror Descent (OMD), based on the Mirror Descent and showed that it is a no-regret algorithm. Optimistic Gradient Descent Ascent (OGDA) is a particular case of OMD, for which the regularizer used is the 2-norm.

In the context of GANs, a first study of OGDA for bilinear games was done by Daskalakis et al. (2018). They proved last-iterate convergence of OGDA, in the sense that for each $\varepsilon > 0$ one can find $\eta > 0$ and T large enough such that (x_T, y_T) (as defined in Algorithm 2) is ε-close to a Nash equilibrium. Later, Liang and Stokes (2019), proved that in the case of a square invertible matrix A, OGDA converges to $(0,0)$ at exponential speed for some specific value of the step size η. Gidel et al. (2019) studied GANs through variational inequalities, with a convergence ratio not better than the one proved in Liang and Stokes (2019).

Another variant of GDA is the Extra-Gradient method (EG) introduced by Korpelevich (1976). Mokhtari et al. (2020a) showed that OGDA and EG can be seen as approximations of the Proximal Point method, and proved the exponential convergence of the three algorithms for square invertible bilinear games and a particular value of η. Zhang and Yu (2020) deeply studied GDA, EG, ODGA, and a heavy ball method. Another related work is the one of Hsieh et al. (2019) who studied the stochastic setting for Extra-Gradient and OGDA.

Beyond bilinear games, several papers considered OGDA for concave/convex games. In particular, Daskalakis and Panageas (2018) studied the stability of fixed points of the dynamics, and Abernethy et al. (2018) improved the rate of convergence for some specific concave/convex games. However the non-concave/non-convex setting exhibits problematic properties: Hsieh et al. (2021) showed there may exist attractors of OGDA containing no stationary points, and Daskalakis et al. (2021) showed that finding an approximate local min-max equilibrium is PPAD-hard. Nevertheless, in a recent paper, Diakonikolas et al. (2021) showed positive results in a new class of non-concave/non-convex min-max problems.

Many studies were also done in compact settings, when $\mathcal{X}$ and $\mathcal{Y}$ are compact. Daskalakis and Panageas (2019) proved the convergence of an Optimistic version of Multiplicative Weight Updates on the simplex, for an appropriate value of the learning rate. Later, Wei et al. (2021) improved their results by showing the convergence for any learning rate smaller than some constant. Finally, in a recent paper Anagnostides et al. (2022) considered the convergence of a projected version of OGDA for general-sum games, where after each iteration, a projection on the simplex is applied.

Notations. We use the following standard notations: $\mathbb{R}_{\leqslant 0}$ is the set of non-positive reals. A^T is the transpose of a matrix A in $\mathbb{C}^{n\times m}$. If A is square, $\mathrm{Sp}(A)$ is the set of complex eigenvalues of A and $\rho(A)=\max\{|\lambda| : \lambda\in \mathrm{Sp}(A)\}$ is the spectral radius of A. Given column vectors x, x' in $\mathbb{R}^n$ or more generally in $\mathbb{C}^n$, we use the Hermitian scalar product $\langle x,x'\rangle = x^T\overline{x'}$ where $\overline{x'}$ is the complex conjugate of x', and the norm $\|x\|=\sqrt{\langle x,x\rangle}$. I_n denotes the identity matrix in $\mathbb{R}^{n\times n}$

3.2 Zero-sum bilinear games

Let A be a payoff matrix in $\mathbb{R}^{n\times m}$, $b\in\mathbb{R}^n$, $c\in\mathbb{R}^m$ and $d\in\mathbb{R}$. We consider the zero-sum game where simultaneously player 1 chooses x in $\mathbb{R}^n$, player 2 chooses y in $\mathbb{R}^m$, and finally player 2 pays $x^TAy+b^Tx+c^Ty+d$ to player 1. Here b, c, x and y are seen as column vectors, and d can w.l.o.g. be set to 0. The set, potentially empty, of Nash equilibria (or saddle points) is $\{(x,y) : A^Tx+c=0, Ay+b=0\}$. Algorithm 2 reduces here to Algorithm 9. Notice that the fixed points of OGDA are exactly the Nash equilibria of the game.

Algorithm 9 Optimistic Gradient Descent Ascent for Zero-sum Games

Input: A matrix A in $\mathbb{R}^{n\times m}$, and vectors $b\in\mathbb{R}^n$, $c\in\mathbb{R}^m$.
Initialization: A step size $\eta>0$, an initialization $(x_0,y_0,x_{-1},y_{-1})\in\mathbb{R}^{(n+m+n+m)}$
for $t=0,\ldots,$ **simultaneously do**
$x_{t+1}=x_t+2\eta(Ay_t+b)-\eta(Ay_{t-1}+b)$
$y_{t+1}=y_t-2\eta(A^Tx_t+c)+\eta(A^Tx_{t-1}+c)$.
end for

3.2.1 Convergence of OGDA for zero-sum games

We show here that as soon as a Nash equilibrium exists, OGDA converges exponentially fast, and we characterize the limit, the speed of convergence and the step sizes where convergence holds.

Theorem 3.1. *Let $A\in\mathbb{R}^{n\times m}\backslash\{0\}$, $b\in\mathbb{R}^n$, $c\in\mathbb{R}^m$, $d\in\mathbb{R}$ and $\eta>0$. We write $\mu_{\max}=\max\{\mu\in\mathbb{R} : \mu\in\mathrm{Sp}(A^TA)\}$, $\mu_{\min}=\min\{\mu>0 : \mu\in\mathrm{Sp}(A^TA)\}$, and consider the Optimistic Gradient Descent Ascent algorithm (Algorithm 9). Fix any initial condition $(x_0,y_0,x_{-1},y_{-1})\in\mathbb{R}^{(n+m+n+m)}$.*

If the set of Nash equilibria is empty, then $(x_t,y_t)_t$ diverges. In the sequel we assume it is not empty, and let D be the distance from (x_0,y_0,x_{-1},y_{-1}) to the set $\{(x,y,x,y)\in\mathbb{R}^{(n+m+n+m)} : A^Tx+c=0, Ay+b=0\}$. For $\eta<\frac{1}{\sqrt{3\mu_{\max}}}$, we have the following:

1) $(x_t,y_t)_t$ converges to the Nash equilibrium (x_∞,y_∞), where x_∞ is the orthogonal projection of x_0 onto $\{x\in\mathbb{R}^n : A^Tx+c=0\}$, and y_∞ is the orthogonal projection of y_0 onto $\{y\in\mathbb{R}^m : Ay+b=0\}$.

2) Define $\lambda(\eta)=\max\{\lambda_*,\lambda_{**}\}$, with

$$\lambda_*=\mathbf{1}_{\mu_{\min}\leqslant\frac{1}{4\eta^2}}\sqrt{\frac{1}{2}(1+\sqrt{1-4\eta^2\mu_{\min}})} \text{ and } \lambda_{**}=\mathbf{1}_{\mu_{\max}\geqslant\frac{1}{4\eta^2}}\sqrt{2\eta^2\mu_{\max}+\eta\sqrt{\mu_{\max}}\sqrt{4\eta^2\mu_{\max}-1}}\,.$$

2a) If $\frac{1}{4\eta^2} \notin \mathrm{Sp}(A^TA)$*, the convergence is exponential with ratio* $\lambda(\eta)$*:*

$$\forall t \geqslant 0, \|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C(\eta)\, D\, \lambda(\eta)^t, \text{ where}$$

$$C(\eta) = \max\{C_*, C_{**}\}, C_* = \mathbf{1}_{\mu_{\min} < \frac{1}{4\eta^2}} \sqrt{\frac{2}{1 - \sqrt{\frac{1+5\eta^2\mu_*}{2+\eta^2\mu_*}}}}, C_{**} = \mathbf{1}_{\mu_{\max} \geqslant \frac{1}{4\eta^2}} \sqrt{\frac{2}{1 - \sqrt{\frac{2+\eta^2\mu_{**}}{1+5\eta^2\mu_{**}}}}},$$

$$\mu_* = \max\{\mu \in \mathrm{Sp}(A^TA) : \eta\sqrt{\mu} < 1/2\} \text{ and } \mu_{**} = \min\{\mu \in \mathrm{Sp}(A^TA) : \eta\sqrt{\mu} > 1/2\}\,.$$

2b) If $\eta \geqslant \frac{1}{2\sqrt{\mu_{\max}}}$ *and* $\frac{1}{4\eta^2} \in \mathrm{Sp}(A^TA)$*, then for every* $\lambda > \lambda(\eta)$ *there exists a constant* C' *such that for all* $t \geqslant 0$*,* $\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C' D \lambda^t$*.*

3) Moreover $\lambda(\eta)$ *is optimal: there exist an initial condition and a constant* $C'' > 0$ *such that* $\|(x_t, y_t) - (x_\infty, y_\infty)\| \geqslant C'' \lambda(\eta)^t$*.*

The proof, in the Appendix, is based on a precise spectral analysis of the linear system with variable $(x_{t+1}, y_{t+1}, x_t, y_t)$, one difficulty being to control the angles between complex eigenspaces in order to obtain proper speeds of convergence. The theorem generalizes all known results on the topic, as shown in Table 1 in the appendix.

3.2.2 Optimal geometric ratio of convergence

Given a matrix $A \neq 0$, one can ask which $\eta \in \left(0, \frac{1}{\sqrt{3\mu_{\max}}}\right)$ minimizes the geometric parameter $\lambda(\eta)$. Computations show that the optimal values $\eta^\star$ and $\lambda^\star = \lambda(\eta^\star)$ for the step size η and the exponential parameter $\lambda(\eta)$ depends on the ratio $\alpha = \frac{\mu_{\min}}{\mu_{\max}} \in (0, 1]$: introducing $\beta = \left(3 + 6\alpha - \alpha^2 - (1-\alpha)\sqrt{(1-\alpha)(9-\alpha)}\right)$, we obtain:

$$\eta^\star \sqrt{\mu_{\max}} = \sqrt{\frac{\beta}{32\alpha}} \in \left[\frac{1}{2}, \frac{1}{2\sqrt{\alpha}}\right] \text{ , and } \lambda^\star = \sqrt{\frac{1}{2}\left(1 + \sqrt{1 - \frac{\beta}{8}}\right)} \geqslant \sqrt{1 - \frac{\alpha}{2}}\,.$$

With this simple formulation[1], we see that when α varies from 0 to 1, $\lambda^\star$ decreases from 1 to $\frac{1}{\sqrt{2}}$: the convergence is faster when α increases, and is optimal when $\alpha = 1$, i.e. when AA^T has a unique non-zero eigenvalue. It is for instance the case when A is, up to a multiplicative constant, an orthogonal matrix. Whatever the value of α, we have $\eta^\star \geqslant \frac{1}{2\sqrt{\mu_{\max}}}$, justifying the importance of the cases where $\eta \geqslant \frac{1}{2\sqrt{\mu_{\max}}}$ in Theorem 3.1.

3.3 General-sum bilinear games

In this section, we extend the analysis of Section 3.2 to the class of general-sum bilinear games. Such a game is now given by two matrices A and B in $\mathbb{R}^{n\times m}$, vectors b, e in $\mathbb{R}^n$, c, f in $\mathbb{R}^m$, and constants d, g in $\mathbb{R}$. Simultaneously player 1 chooses x in $\mathbb{R}^n$ and player 2 chooses y in $\mathbb{R}^m$. Then, the payoff for player 1 is $g_1(x, y) = x^TAy + b^Tx + c^Ty + d$, and the

[1] In Zhang and Yu (2020), complex expressions for $\eta^\star$ and λ^* are given, however their approach relies on a incorrect statement (Theorem 2.1 there), where no attention is paid to the eigenvalue 1.

payoff for player 2 is $g_2(x,y) = x^T By + e^T x + f^T y + g$. The set of Nash equilibria is now $\{(x,y) \in \mathbb{R}^{(n+m)} : B^T x + f = 0, Ay + b = 0\}$. In Section 3.2 we have considered the zero-sum case where $(B,e,f,g) = -(A,b,c,d)$.

We now introduce OGDA for general-sum games in Algorithm 10. Without loss of generality, d and g can be set to 0. Notice that if $(x_t,y_t)_t$ converges, the limit is a Nash

Algorithm 10 Optimistic Gradient Descent Ascent for General-sum Games

Input: Two matrices A and B in $\mathbb{R}^{n\times m}$, and vectors b, e in $\mathbb{R}^n$, c, f in $\mathbb{R}^m$.
Initialization: A step size $\eta > 0$, an initialization $(x_0, y_0, x_{-1}, y_{-1}) \in \mathbb{R}^{(n+m+n+m)}$
for $t = 0, \ldots,$ **simultaneously do**
$x_{t+1} = x_t + 2\eta(Ay_t + b) - \eta(Ay_{t-1} + b)$
$y_{t+1} = y_t + 2\eta(B^T x_t + f) - \eta(B^T x_{t-1} + f)$.
end for

equilibrium. The first result here is a counterexample to the convergence of OGDA.

Proposition 3.2. *OGDA may fail to converge for some general sum-games.*

Proof. A simple case is when $A = B = (1)$ and $b = c = d = e = f = g = 0$. Assume for simplicity that $x_0 = y_0$ and $x_{-1} = y_{-1}$. Then for all t, $x_t = y_t$ and $x_{t+1} = x_t(1+2\eta) - \eta x_{t-1}$. This can be written $x_{t+1} - x_t = \eta x_t + \eta(x_t - x_{t-1})$, so that if $x_0 > \max\{x_{-1}, 0\}$ then $x_{t+1} > \max\{x_t, 0\}$ for all t. It implies $x_{t+1} > x_t(1+\eta)$ for all t, so $x_t \xrightarrow[t\to\infty]{} +\infty$. □

3.3.1 Sufficient conditions for convergence in general-sum games

We now provide sufficient conditions for convergence.

Theorem 3.3. *Let A, B be in $\mathbb{R}^{n\times m}$ with $\mathrm{Sp}(B^T A) \subset \mathbb{R}_{\leqslant 0}$, b, e in $\mathbb{R}^n$ and c, f in $\mathbb{R}^m$. Let $\eta \in (0, \frac{1}{2\sqrt{\mu_{\max}}})$, where $\mu_{\max} = \rho(B^T A) = \rho(AB^T)$ is the largest eigenvalue of $B^T A$. Assume either that A and B are square invertible matrices or that $B^T A$ is diagonalizable, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$ and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$. Moreover, assume that a Nash equilibrium exists, i.e. that $\{(x,y) \in \mathbb{R}^{(n+m)} : B^T x + f = 0, Ay + b = 0\} \neq \varnothing$.*

Given an initialization $(x_0, y_0, x_{-1}, y_{-1}) \in \mathbb{R}^{(n+m+n+m)}$, consider the Optimistic Gradient Descent Algorithm 10. Then:

1) $(x_t, y_t)_t$ converges to a Nash equilibrium (x_∞, y_∞).

2) x_∞ is the linear projection of x_0 onto $\{x \in \mathbb{R}^n : B^T x + f = 0\}$ along $(\mathrm{Ker}(A^T))^\perp = \mathrm{Im}(A)$ and y_∞ is the linear projection of y_0 onto $\{y \in \mathbb{R}^m : Ay + b = 0\}$ along $(\mathrm{Ker}(B))^\perp = \mathrm{Im}(B^T)$.

3) The convergence is exponential: there exists a constant $C > 0$ verifying for all $t \geqslant 0$,

$$\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C \, \|(x_0, y_0, x_{-1}, y_{-1})\| \, \lambda(\eta)^t \,,$$

where $\lambda(\eta) = \sqrt{\frac{1}{2}(1 + \sqrt{1 - 4\eta^2 \mu_{\min}})}$, with $\mu_{\min} = \min\{ \mu > 0 : -\mu \in \mathrm{Sp}(B^T A)\}$ and the convention $\lambda(\eta) = 0$ if $A = B = 0$.

To highlight the assumptions of Theorem 3.3, consider for instance the case where $A = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Then $B^T A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ is diagonalizable, and its sole eigenvalues 0 belongs to $\mathbb{R}_{\leqslant 0}$. However, for the initial conditions $x_0 = x_{-1} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $y_0 = y_{-1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, we get for any $t \geqslant 0$ that $x_t = \begin{pmatrix} 1+\eta t \\ -1-\eta t \end{pmatrix}$ and $y_t = \begin{pmatrix} 0,1 \end{pmatrix}$, which does not converge to a Nash equilibrium of the game.
This is not in contradiction with Theorem 3.3 since here the condition $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^2$ is not verified.

3.3.2 Improving the zero-sum rate of convergence with general-sum games

Consider a zero-sum game $g(x,y) = x^T A y + b^T x + c^T y$ with $A \neq 0$, and assume we are interested in finding an optimal strategy y for player 2. We have seen in Section 3.2.2 that the optimal geometric parameter $\lambda(\eta)$ is at least $\sqrt{1-\frac{\alpha}{2}} \geqslant 1/\sqrt{2}$, where $\alpha = \frac{\mu_{\min}(AA^T)}{\mu_{\max}(AA^T)}$. We now show how to use general-sum games to obtain the optimal speed of convergence of $1/\sqrt{2}$.

Let us look at the convergence ratios in Theorem 3.3. For η optimally chosen close to $\frac{1}{2\sqrt{\mu_{\max}}}$, we get a rate of $\lambda(\eta^\star) = \sqrt{\frac{1}{2}\left(1+\sqrt{1-\frac{\mu_{\min}}{\mu_{\max}}}\right)}$, which is minimized when $\mu_{\min} = \mu_{\max}$, that is when $B^T A$ has a unique nonzero eigenvalue.

We introduce $B = -(A^\dagger)^T$, where $A^\dagger$ is the Moore-Penrose inverse of A. For instance if $A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}$ we have $A^\dagger = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix}$ and put $B = \begin{pmatrix} -1 & 0 \\ 0 & -\frac{1}{2} \\ 0 & 0 \end{pmatrix}$. We can show that the assumptions of Theorem 3.3 are satisfied and obtain:

Theorem 3.4. *Let $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, and $0 < \eta \leqslant \frac{1}{2}$. Consider the Optimistic Gradient Descent Ascent 10 for the payoffs $g_1(x,y) = x^T A y + b^T x + c^T y$ and $g_2(x,y) = x^T B y + e^T x + f^T y$ with $B = -(A^\dagger)^T$ and some $e \in \mathbb{R}^n$, $f \in \mathbb{R}^m$. We assume the existence of a Nash equilibrium, i.e. of x, y such that $A^\dagger x = f$ and $Ay + b = 0$.*

Then $(y_t)_t$ converges to the orthogonal projection of y_0 onto $\{y \in \mathbb{R}^m : Ay + b = 0\}$, and $(x_t)_t$ converges to the orthogonal projection of x_0 onto $\{x \in \mathbb{R}^n : A^\dagger x = f\}$.

Moreover, the convergence is exponential: if $\eta < 1/2$ there exists a constant $C > 0$ s.t. for all $t \geqslant 0$, $\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C \, \|(x_0, y_0, x_{-1}, y_{-1})\| \, \lambda(\eta)^t$, where

$$\lambda(\eta) = \sqrt{\frac{1}{2}(1+\sqrt{1-4\eta^2})}\,.$$

And if $\eta = 1/2$ then for every $\lambda > 1/\sqrt{2}$ there exists $C > 0$ s.t. for all $t \geqslant 0$, $\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C \, \|(x_0, y_0, x_{-1}, y_{-1})\| \, \lambda^t$.

Here also, $\lambda(\eta)$ can be shown to be the exact rate for exponential convergence, and if $\eta \simeq 1/2$ we obtain the fully optimal rate of $\frac{1}{\sqrt{2}}$. Notice that the limit for $(y_t)_t$ is the same

as in Theorem 3.1 when $B = -A$, but the speed of convergence is much better when we use the OGDA algorithm in this general-sum context with $B = -(A^\dagger)^T$.

Notice also that if $f = A^\dagger x^*$ for some x^* satisfying $A^T x^* + c = 0$, then $(x_t)_t$ also converges to the same limit as in Theorem 3.1. In particular if $c = 0$, one can choose $f = 0$ to ensure this property.

3.3.3 "Cooperation" induced by OGDA

Assume the game $(x^T Ay, x^T By)$ has a "potential for cooperation", in the specific sense that there exist actions $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ such that both $x^T Ay$ and $x^T By$ are positive. Define the sequence $(x_t, y_t)_t$ by $(x_t, y_t) = t(x, y)$ for each t, then both $x_t^T Ay_t \xrightarrow[t\to+\infty]{} +\infty$ and $x_t^T By_t \xrightarrow[t\to+\infty]{} +\infty$. Even though we are not reaching a Nash equilibrium, this may be seen as a desirable outcome of the interaction between the players, who aim at maximizing their payoffs. A particular case of game with such cooperation is common-payoff games, for which $A = B$.

We now present our last theorem, related to the convergence of OGDA to a Nash equilibrium or to infinite payoffs in the case of general general-sum games with payoffs $(x^T Ay + b^T x + c^T y + d, x^T By + e^T x + f^T y + g)$:

Theorem 3.5. *Let A, B be in $\mathbb{R}^{n\times m}$, b, e in $\mathbb{R}^n$, c, f in $\mathbb{R}^m$, and d, g in $\mathbb{R}$, with $\mathrm{Sp}(B^T A) \subset \mathbb{R}$. Let $\eta \in (0, \frac{1}{2\sqrt{\mu_{\max}}})$, where $\mu_{\max} = \rho(B^T A) = \rho(AB^T)$ is the largest eigenvalue of $B^T A$. Assume that $B^T A$ is diagonalizable, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$, $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$, and that a Nash equilibrium exists, i.e. that $\{(x, y) \in \mathbb{R}^{(n+m)} : B^T x + f = 0, Ay + b = 0\} \neq \varnothing$.*

Given an initialization $(x_0, y_0, x_{-1}, y_{-1}) \in \mathbb{R}^{(n+m+n+m)}$, consider the Optimistic Gradient Descent Ascent (Algorithm 10). Then either $(x_t, y_t)_t$ converges exponentially fast to a Nash equilibrium, or the current payoffs $x_t^T Ay_t + b^T x_t + c^T y_t + d$ and $x_t^T By_t + e^T x_t + f^T y_t + g$ converge exponentially fast to $+\infty$.

The assumptions $B^T A$ diagonalizable with all eigenvalues in $\mathbb{R}$, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$ and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$ are in particular true in each of the following cases:

1) $B = \alpha A$ for some real number $\alpha \neq 0$,

2) A and B are square matrices in $\mathbb{R}^{n\times n}$ such that $B^T A$ is diagonalizable with n distinct non-zero real eigenvalues.

Case *1*) includes the case of common payoffs games $A = B$, as well as the case of zero-sum games $B = -A$.

The theorem does not extend to all pairs (A, B) of matrices. For instance if $A = \begin{pmatrix} 3 & 8 \\ -2 & 3 \end{pmatrix}$ and $B = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$. Then $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^2$, $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^2$, and $B^T A$ is diagonalizable, but with complex nonreal eigenvalues. It can be shown that for some initial conditions, OGDA does not converge to a Nash equilibrium, and that the payoffs for both players do not converge to $+\infty$. This comes from the fact that $B^T A$ has complex eigenvalues, not respecting the assumptions of the theorem. This result is summarized in Proposition 3.6. The details of the proof are in the appendix.

Proposition 3.6. *There exists some general-sum games (A,B) with $A,B \in \mathbb{R}^{2\times 2}$, an initialization vector (x_0,y_0,x_{-1},y_{-1}) and a learning rate $0<\eta<\frac{1}{2\sqrt{\mu_{\max}}}$, such that, denoting (x_t,y_t) the generated sequence by OGDA, we have:*
For all $M>0$ and $T\in\mathbb{N}$, there exist $t_1,t_2>T$ with $x_{t_1}^T A y_{t_1} > M$ and $x_{t_2}^T A y_{t_2} < -M$.

It shows that there exist a game and a run of OGDA on this game such that the payoff of player 1 neither converges, nor tends towards $+\infty$.

3.4 Illustration: Generative Adversarial Networks

3.4.1 For bilinear games

Generative Adversarial Networks (GANs) have been introduced in Goodfellow et al. (2014). The Wasserstein variant (WGANs), introduced in Arjovsky et al. (2017), is based on the Wasserstein distance $W_1(\nu,\nu') = \sup_{f\in \mathrm{Lip}_1} \mathbb{E}_\nu(f) - \mathbb{E}_{\nu'}(f)$ between probability measures ν and ν'. A deep theoretical study is of the convergence of WGANs under some hypotheses is done in Biau et al. (2021). We illustrate here our results on a simple example of WGANs derived from Daskalakis et al. (2018).

We assume that the data follows a multivariate normal distribution $\nu^* = \mathcal{N}(v, I_n)$, where v is unknown. The generator chooses θ in $\mathbb{R}^n$ and generates the distribution $G_\theta(z) = \theta + A_1 z$, where $A_1 \in \mathbb{R}^{d\times m}$ is a fixed matrix and $z\sim\mathcal{N}(0,I_m)$ is the noise. The goal of the generator is to find θ minimizing the Wasserstein distance between the laws of $G_\theta(z)$ and ν^*. For the set of discriminators, instead of considering all 1-Lipschitz functions we use the toy set of all $D_\beta : x \mapsto \langle A_2\beta, x\rangle$, where $\|\beta\|_2 \leqslant 1$ and A_2 is a fixed matrix in $\mathbb{R}^{n\times n}$ satisfying $\rho(A_2^T A_2) \leqslant 1$ so that each D_β is 1-Lipschitz. We assume to have access to true expectations rather than samples of data, and the WGAN problem becomes:

$$\inf_\theta \sup_\beta \left(\mathbb{E}_{x\sim\mathcal{N}(v,I_d)} \langle A_2\beta, x\rangle - \mathbb{E}_{z\sim\mathcal{N}(0,I_m)} \langle A_2\beta, \theta + A_1 z\rangle \right) ,$$

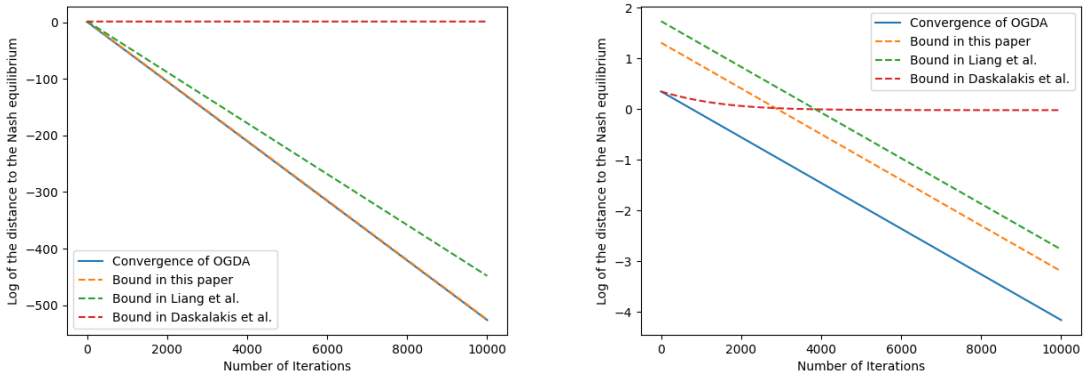
that is

$$\inf_\theta \sup_\beta \langle A_2\beta, v-\theta\rangle .$$

We are in the case $x^T A y + b^T x + c^T y + d$ of Section 3.2 with $x=\beta$, $y=\theta$, $A=-A_2^T$, $b=A_2^T v$, $c=0$ and $d=0$. θ is arbitrary in $\mathbb{R}^n$. A priori we have the constraint $\|\beta\|_2\leqslant 1$, but allowing arbitrary $\|\beta\|$ does not change the value nor the optimal strategies of the generator, so we simply skip the constraint and consider any β in $\mathbb{R}^n$. The set of Nash equilibria is then $\mathrm{Ker}(A_2)\times\{\theta\in\mathbb{R}^n : v-\theta\in\mathrm{Ker}(A_2^T)\}$.

With these settings, Theorem 3.1 implies that for $\eta<\frac{1}{2}$, OGDA will converge exponentially fast, with ratio $\lambda(\eta) = \sqrt{\frac{1}{2}(1+\sqrt{1-4\eta^2\mu_{\min}})}$, and some multiplicative constant $C(\eta)$.

We have considered the case $n=2$, $A_1=A_2=I_2$, and $v=(3,4)$. In Figure 3.1 the log-distance of the iterate given by OGDA to the set of Nash equilibria is plotted as a function of the number of iterations. This confirms the exponential convergence of OGDA for different values of η, and shows the optimality up to a multiplicative constant of the



(a) $\eta = 0.3$ (b) $\eta = 0.03$

Figure 3.1: Log plot of the distance to the Nash equilibrium. The red line shows the speed of convergence of OGDA and each dashed line shows the different bounds: green for Daskalakis et al. (2018), orange for Liang and Stokes (2019) and blue for ours.

bound given in Theorem 3.1. The plots also illustrate the decrease of the geometric ratio $\lambda(\eta)$ with η as long as $\eta \leqslant \eta^\star$. Note that the two values of η given here are smaller than $\frac{1}{2\sqrt{\mu_{\max}}}$, for a better comparison with previous papers, but we showed that OGDA converges for any $\eta \leqslant \frac{1}{\sqrt{3\mu_{\max}}}$.

We now consider the speed improvement we presented in Subsection 3.3.2. Assume $A_2^T \neq 0$ and recall that $\mu_{\max} = \max\{\mu : \mu \in \mathrm{Sp}(A_2^T A_2)\} \leqslant 1$, $\mu_{\min} = \min\{\mu > 0 : \mu \in \mathrm{Sp}(A_2^T A_2)\}$ and $\alpha = \frac{\mu_{\min}}{\mu_{\max}}$. Running OGDA with $\eta < \frac{1}{\sqrt{3\mu_{\max}}}$, Theorem 3.1 ensures that (β_t) converges to a Nash equilibrium of the game. Choosing the best step size $\eta^\star$, the optimal ratio of convergence $\lambda(\eta^\star)$ is greater than $\sqrt{1-\frac{\alpha}{2}}$ as seen in Section 3.2.2.
Let us introduce $B = A_2^\dagger$ and run OGDA for the general-sum game where $g_1(x,y) = x^T(-A_2^T)y + v^T A_2 x$ and $g_2(x,y) = x^T B y$). According to Theorem 3.3, OGDA now converges to the same limit for both x_t and y_t), but with minimum geometric ratio $\lambda'(1/2) = \frac{1}{\sqrt{2}} < \sqrt{1-\frac{\alpha}{2}}$, improving the previous rate of convergence.

This gain of speed can be seen on Figure 3.2 where $-A_2 = A = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}$ and $v = (1,1)$. The convergence rate goes from $\lambda(\eta^\star) \simeq 0.9538$ for the optimal rate $\eta^\star \simeq 0.5608$ for zero-sum games, to $\lambda'(1/2) \simeq 0.7071$ with our general-sum improvement. Using a general-sum game with the matrix $A_2^\dagger$ greatly improves the speed of convergence when α is not close to 1.

Now that we studied the results for the bilinear case, one could ask whether something similar can be done to the concave/convex case.

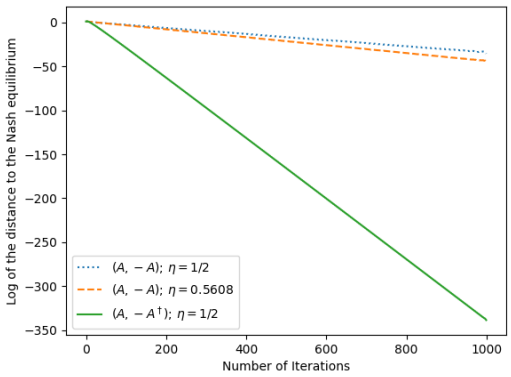


Figure 3.2: Comparison of the speed of convergence of OGDA with different variants. Standard OGDA with $\eta = 1/2$ is in blue, standard OGDA with the best η is in orange, and OGDA with general sum-game using $A^\dagger$ is in green.

3.4.2 Experimental results for some concave/convex games

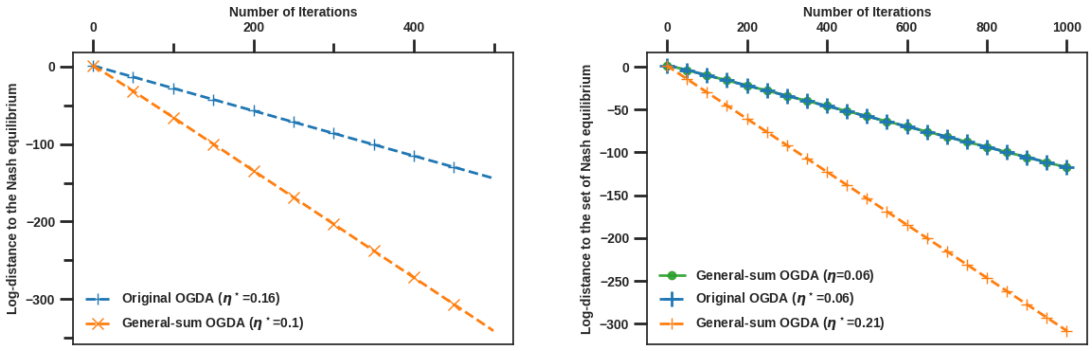
In the concave/convex case where $g(x,y)$ is concave in x and convex in y, Daskalakis and Panageas (2018) proved the convergence of OGDA to a Nash equilibrium under some stability hypotheses, but in general the best convergence rate is not known. We present here how our general-sum improvement of the convergence rate can be used, in the particular case where the function g can be written $g(x,y) = \psi(x)^T A\phi(y)$, with ψ, ϕ smooth, and g concave in x and convex in y. These sort of games are called hidden bilinear games, and were introduced in Vlatakis-Gkaragkounis et al. (2019). We can use Algorithm 2 and run OGDA, but we can also run the general-sum extension of OGDA to the game with payoff function $g_1 = g$ for player 1, and $g_2(x,y) = -\psi(x)^T(A^\dagger)^T\phi(y)$ for player 2. Figure 3.3 below shows how our general-sum variant improves the speed of convergence to equilibria in the following 2 examples.

In example a), $g(x,y) = -2x^2 + \frac{1}{3}y^2 + xy$ for x, y in $\mathbb{R}$. Here $\psi(x) = (-x^2, x, 1)^T$ and $\phi(y) = (y^2, y, 1)^T$, $A = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 1 & 0 \\ \frac{1}{3} & 0 & 0 \end{pmatrix}$ so that $A^\dagger = \begin{pmatrix} 0 & 0 & 3 \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}$ and $g_2(x,y) = \frac{1}{2}x^2 - xy - 3y^2$. In both games the unique Nash eq. is $(0,0)$ and we plot the log-distance from the current iteration to the Nash equilibrium for $\eta = 0.06$, and initialized at $x_0 = x_{-1} = 2$ and $y_0 = y_{-1} = -1$.

In example b), x and y belong to $\mathbb{R}^2$, and $g(x,y) = x_1y_1^2 + 3x_2y_2^2 = x^T A\phi(y)$ with $A = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$ and $\phi(y) = (y_1^2, y_2^2)^T$. Here $g_2(x,y) = -x_1y_1^2 - \frac{1}{3}x_2y_2^2$, and the set of Nash equilibria is $\{(x,y), x \geqslant 0, y = 0\}$ in both games. We plot the log-distance from the current iteration to the set of equilibria for both games, for two values of η: 0.06, which is the empirical best value for the standard OGDA, and 0.21, which is the empirical best value for OGDA with the general-sum transformation. The initialization for this plot is

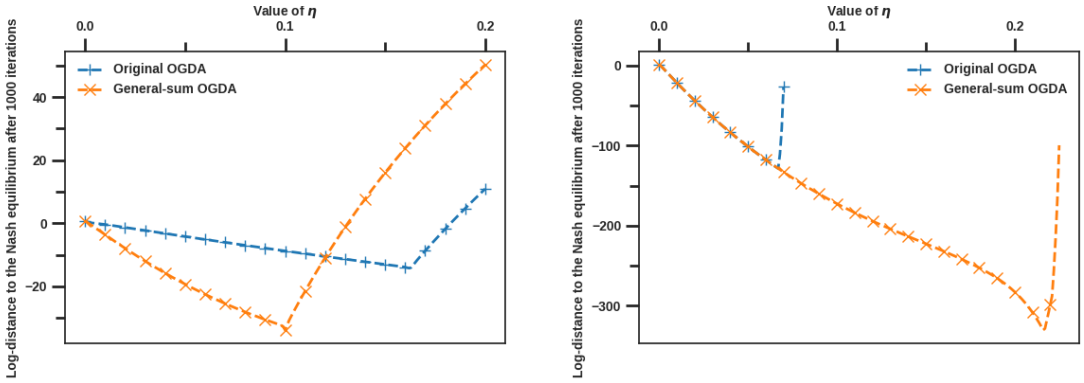
$x_0 = x_{-1} = (-1, 1)^T$ and $y_0 = y_{-1} = (1, -2)^T$. Here for $\eta = 0.21$ the standard OGDA does not converge, and for $\eta = 0.06$ the convergence is similar in both cases.

In our simulations (see Figure 3.4), the best (i.e. optimized in the step size) speed of convergence for the general-sum improvement is always experimentally at least as good, and often significantly better, than the best convergence speed for the original OGDA.



(a) (b)

Figure 3.3: Speed of convergence of OGDA for diverse methods on examples a) and b)



(a) (b)

Figure 3.4: Log-distance to the set of Nash equilibrium w.r.t. η for different methods on examples a) and b)

3.5 Conclusion

We have proved the exponential convergence of OGDA to a Nash equilibrium for any zero-sum bilinear game, and also provided a new, optimal, geometric ratio for convergence. This implies an important stability property for the OGDA system as the number of steps increases. We also identify the limit as a function of the initialization, fully characterize the step sizes inducing convergence, and recover the optimal step size leading to the fastest

possible convergence of OGDA in the zero-sum case. In particular we have obtained that the optimal ratio of convergence is decreasing in the ratio $\alpha = \frac{\mu_{\min}(AA^T)}{\mu_{\max}(AA^T)}$. These results for zero-sum games clarify and generalize the previous results of the literature on the topic.

We have also extended the study to general-sum games, introducing OGDA for such games. We gave sufficient conditions for OGDA to converge to a Nash equilibrium, and showed how to increase the speed of convergence of a min-max problem involving a payoff matrix A by introducing a general-sum game using the matrix $-(A^\dagger)^T$, i.e. minus the transpose of the Moore-Penrose inverse of A. This may be seen as a proof of concept that general-sum games can be used to solve efficiently min-max or zero-sum problems.

Extending this approach to general, non necessarily bilinear, games, may be promising and is left for future research. For instance we know from Mokhtari et al. (2020b) that OGDA converges in convex/concave games under some hypotheses, and a first step would be to try to adapt our approach to their setup.

We also showed that in an important class of games (including the common payoffs case as well as the zero-sum case) either OGDA converges to a Nash equilibrium, or the payoffs of both players converge to $+\infty$.

Bilinear games are widely used as a toy model for many applications, and we illustrated our results on a GAN example. This may be used as an additional argument in favor of using the Optimistic version of GDA in GANs. Nevertheless, if one wants to go further in the application to GANs, there are several factors that should be taken into account. One of them would be the stochasticity of the payoffs. Some experiments were made in Daskalakis et al. (2018), and some theoretical bounds were given in Hsieh et al. (2019), but a lot still needs to be done. A second and harder factor is the non-concave/non-convex properties of real instances of GANs.

All in all, the general comprehension of OGDA is improving, and so are the applications to the stability of first-order methods for GANs. This paper aims at continuing this dynamics, in giving more insights on zero-sum and general-sum problems.

3.A Comments on Theorem 3.1

Several remarks are in order regarding Theorem 3.1:

a) We know from Theorem 3.1 that the upper bound $\lambda(\eta)$ we give on the exponential parameter matches the lower bound that we found.

The optimality of $\lambda(\eta)$ shows that our result for zero-sum games improves the existing literature. We still do a summary in Table 3.1 below of the main related papers, to see how our theorem compares with their results.

Table 3.1: Comparison table of the convergence rate

Reference	Convergence Speed	Geometric Ratio w.r.t η	Constant	Step Size
Daskalakis et al. (2018)	Polynomial	✗	✗	$\eta < \frac{\mu_{\min}^2}{3}$
Liang and Stokes (2019)	Exponential	$\exp(\frac{-\eta^2\mu_{\min}}{2})$	4	$\eta = \frac{1}{\sqrt{8\mu_{\max}}}$
Peng et al. (2020)	Exponential	$\sqrt{\frac{1}{2}\left(1+\sqrt{1-\eta^2\mu_{\min}}\right)}$	✗	$\eta \leqslant \frac{1}{2\sqrt{\mu_{\max}}}$
Mokhtari et al. (2020a)	Exponential	$\sqrt{1-\frac{\mu_{\min}\eta^2}{4}}$	1	$\eta = \frac{1}{\sqrt{40\mu_{\max}}}$
Zhang and Yu (2020)	Exponential	✗[2]	✗	$\eta < \frac{1}{\sqrt{3\mu_{\max}}}$
This paper	Exponential	$\sqrt{\frac{1}{2}\left(1+\sqrt{1-4\eta^2\mu_{\min}}\right)}$	$C(\eta) = \sqrt{\frac{2}{1-\sqrt{\frac{1+5\eta^2\mu_{\max}}{2+\eta^2\mu_{\max}}}}}$	$\eta < \frac{1}{2\sqrt{\mu_{\max}}}$
This paper	Exponential	$\lambda(\eta^\star)$ (optimal result)	$C(\eta)$	$\eta < \frac{1}{\sqrt{3\mu_{\max}}}$

b) Theorem 3.1 also holds for $A = 0$ if we use the convention $\lambda(\eta) = 0$ in this case.

c) Both $\lambda_*(\eta)$ and $\lambda_{**}(\eta)$ are in $[0,1)$ and $\lambda(\eta) \in [\frac{\sqrt{2}}{2}, 1)$.

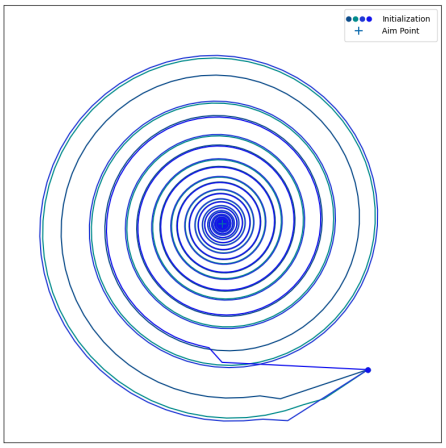
d) Notice that the speed of convergence is independent of the dimension: $\lambda(\eta)$, as well as $C(\eta)$ and D from part 2) of the Theorem only depends on $\mathcal{S}(A)$ and η, and not on n nor p.

e) The step-size η is fixed, and does not evolve with time. If $\eta < \frac{1}{2\sqrt{\mu_{\max}}}$, the ratio of convergence $\lambda(\eta)$ is better when η is large, so that having η too small is not a good option for large t. We elaborate on the optimal choice of η in Subsection 3.2.2.

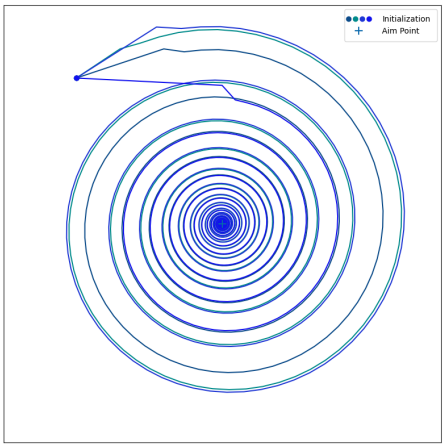
f) We finally illustrate a part of Theorem 3.1: the expression of the limit value (x_∞, y_∞) of OGDA with respect to the initialization only depends on x_0 and y_0, but not on x_{-1} and y_{-1}. For this, we plotted in Figure 3.5 several runs of OGDA for the game with matrix $A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. On the left image, we plotted the first coordinate of x_t with the first coordinate of y_t, and on the right image, we plotted the second coordinate of x_t with the second coordinate of y_t. For each of the run, the same vector (x_0, y_0) was chosen, and different values were taken for x_{-1} and y_{-1}. We

[2] However this paper contains an expression for the ratio $\lambda(\eta^\star)$ with optimal step size.

can see that changing the vectors of time -1 changes the dynamic of the system, but that all of the experiments still converge to the same target point, which only depends on the vector (x_0, y_0), as one can see from Theorem 3.1.



(a) $x_{t,1}$ with respect to $y_{t,1}$



(b) $x_{t,2}$ with respect to $y_{t,2}$

Figure 3.5: Several plots of OGDA

3.B Proof of Theorem 3.1

Let us fix $A \in \mathbb{R}^{n\times m} \neq 0$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$, $d \in \mathbb{R}$ and $\eta > 0$.

Let us also introduce some new notations, and define some variables that will be of great importance in the following:

Definition 3.7. We denote by $\mathcal{S}(A)$ the set $\mathrm{Sp}(A^TA) \cup \mathrm{Sp}(AA^T)$ (recall that A^TA and AA^T have the same non-zero eigenvalues).

Furthermore, let us define the matrix Λ and the vector u by

$$\Lambda = \begin{pmatrix} I_n & 2\eta A & 0 & -\eta A \\ -2\eta A^T & I_m & \eta A^T & 0 \\ I_n & 0 & 0 & 0 \\ 0 & I_m & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+m+n+m)\times(n+m+n+m)}, \; u = \begin{pmatrix} \eta b \\ -\eta c \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^{(n+m+n+m)},$$

and for $t \geqslant 0$, let Z_t be the column vector $Z_t = (x_t, y_t, x_{t-1}, y_{t-1}) \in \mathbb{R}^{(n+m+n+m)}$.

With this definition, we can now see OGDA as a dynamical system in $\mathbb{R}^{(n+m+n+m)}$, with $Z_0 \in \mathbb{R}^{(n+m+n+m)}$ and $\forall t \geqslant 0, Z_{t+1} = \Lambda Z_t + u$. In the sequel we will denote by I the identity matrix of size $2(n+m)$.

For $b=0$, $c=0$ and $d=0$, vector u equals 0, and thus $Z_{t+1} = \Lambda Z_t = \Lambda^{t+1} Z_0$. Then, the set of Nash equilibria is never empty, because it contains the vector $(0,0)$, and the study of this dynamical system depends only on the eigenvalues of Λ.

We divided the proof of Theorem 3.1 in several parts. For the first parts, we assumed that $b=0$, $c=0$, $d=0$.
In Sections 3.B.1, 3.B.2, and 3.B.3 we prove the convergence of OGDA to a Nash equilibrium, first in the case 2a), then in the case 2b) of Theorem 3.1. We explicit the origin of $\lambda(\eta)$, and why it has this dependence in the eigenvalues of AA^T. Later on, in Section 3.B.4 , we study the speed of convergence of OGDA, where we mainly give an upper bound of the constants $C(\mu)$ and D. In the next part (Section 3.B.5), we use the previous results to extend them to any value of $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$ and $d \in \mathbb{R}$. In a last section (Section 3.B.6), we show the optimality of $\lambda(\eta)$, in providing an upper bound that matches our lower bound.

3.B.1 Convergence to a Nash equilibrium for general $\eta > 0$

Let η be a non-negative step size.
In order to express the eigenvalues of Λ, we introduce the following sets:

Definition 3.8. Given $\mu \in \mathbb{C}$, we define

$$S(\mu) = \{\lambda \in \mathbb{C} : \lambda^2(1-\lambda)^2 + \mu\eta^2(1-2\lambda)^2 = 0\}\,.$$

Notice that if $\eta = 0$ or $\mu = 0$, then $S(\mu) = \{0,1\}$. Simple computations imply the following lemma.

Lemma 3.9. *Assume μ is real non-negative. We write $\theta = \eta\sqrt{\mu} \geqslant 0$, and let $\delta \in \mathbb{C}$ be such that $\delta^2 = 1 - 4\theta^2$. More precisely, if $4\theta^2 \leqslant 1$ we use $\delta = \sqrt{1-4\theta^2}$ and if $4\theta^2 \geqslant 1$ we use $\delta = i\sqrt{4\theta^2 - 1}$.*

Then we can write

$$S(\mu) = \{\lambda_1(\mu), \lambda_2(\mu), \lambda_3(\mu), \lambda_4(\mu)\},\ \textit{with}$$

$$\lambda_1(\mu) = \frac{1}{2}(1+\delta+2i\theta) = \overline{\lambda_3}(\mu),\ \textit{and}\ \lambda_2(\mu) = \frac{1}{2}(1-\delta+2i\theta) = \overline{\lambda_4}(\mu).$$

0) If $\theta = 0$, then $S(\mu) = \{0,1\}$.
1) If $0 < \theta < 1/2$, $S(\mu)$ has 4 elements and $|\lambda_1(\mu)|^2 = |\lambda_3(\mu)|^2 = \frac{1}{2}(1+\sqrt{1-4\theta^2}) < 1$, and $|\lambda_2(\mu)|^2 = |\lambda_4(\mu)|^2 = \frac{1}{2}(1-\sqrt{1-4\theta^2}) < \frac{1}{2}$.
2) If $\theta = 1/2$, $S(\mu) = \{\frac{1}{2}(1+i), \frac{1}{2}(1-i)\}$.
3) If $\theta > 1/2$, $S(\mu)$ has 4 elements and $|\lambda_1(\mu)|^2 = |\lambda_3(\mu)|^2 = 2\theta^2 + \theta\sqrt{4\theta^2-1}$, and $|\lambda_2(\mu)|^2 = |\lambda_4(\mu)|^2 = 2\theta^2 - \theta\sqrt{4\theta^2-1}$.

Notice that for $\theta \geqslant 1/2$, the property $2\theta^2 + \theta\sqrt{4\theta^2-1} < 1$ is equivalent to $\theta < \frac{1}{\sqrt{3}}$.

The link between the eigenvalues of the matrices A^TA and AA^T, and the eigenvalues of Λ is the following.

Lemma 3.10. *The spectrum of Λ is the union of the $S(\mu)$ for μ an eigenvalue of A:*

$$\mathrm{Sp}(\Lambda) = \bigcup_{\mu \in \mathcal{S}(A)} S(\mu)\,.$$

Proof. For $\lambda \in \mathbb{C}$ and $Z = (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)}$, we have:

$$\Lambda Z = \lambda Z \iff \begin{cases} x + 2\eta A y - \eta A y' = \lambda x \\ y - 2\eta A^T x + \eta A^T x' = \lambda y \\ x = \lambda x' \\ y = \lambda y' \end{cases}$$
$$\iff \begin{cases} \lambda(1-\lambda)x' + (2\lambda - 1)\eta A y' = 0 \\ \lambda(1-\lambda)y' - (2\lambda - 1)\eta A^T x' = 0 \\ x = \lambda x' \\ y = \lambda y'\,. \end{cases}$$

a) Assume $\Lambda Z = \lambda Z$ and $Z \neq 0$. Multiplying the first line by $(2\lambda - 1)\eta A^T$ and the second line by $(2\lambda - 1)\eta A$, we get:

$$\begin{cases} \lambda^2(1-\lambda)^2 y' + (2\lambda-1)^2\eta^2 A^T A y' = 0 \\ \lambda^2(1-\lambda)^2 x' + (2\lambda-1)^2\eta^2 A A^T x' = 0\,. \end{cases} \tag{3.2}$$

Since $Z \neq 0$, we have $x' \neq 0$ or $y' \neq 0$, and $\lambda \neq 1/2$. This implies that $-\frac{\lambda^2(1-\lambda)^2}{(2\lambda-1)^2\eta^2}$ is an eigenvalue of $A^T A$ or of AA^T. If we denote by μ such eigenvalue, then $\lambda \in S(\mu)$.

b) Conversely, let μ be an eigenvalue of $A^T A$, and consider $\lambda \in S(\mu)$. Let $x' \neq 0$ be such that $AA^T x' = \mu x'$ and consider y' satisfying:

$$\lambda(1-\lambda)y' = \eta(2\lambda - 1)A^T x'\,.$$

If $\mu \neq 0$, then $\lambda \notin \{0, 1\}$ and y is uniquely defined; if $\mu = 0$ then $AA^T x' = 0$ so $A^T x' = 0$ and any y' will do. In each case, one can check that $\Lambda(\lambda x', \lambda y', x', y')^T = \lambda Z$, so that λ is an eigenvalue of Λ. This also stands for μ an eigenvalue of AA^T.

a) and b) show that the set of eigenvalues of Λ is $\{\lambda : \lambda \in S(\mu), \mu \in \mathcal{S}(A)\}$. □

Combining Lemma 3.9 and Lemma 3.10 yields;

Corollary 3.11.

$$\max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda)\backslash\{1\}\} < 1 \iff \forall \mu \in \mathcal{S}(A), \eta\sqrt{\mu} < \frac{1}{\sqrt{3}} \iff \eta < \frac{1}{\sqrt{3\mu_{\max}}}\,.$$

And if $\eta > \frac{1}{\sqrt{3\mu_{\max}}}$ we have $\max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda)\} > 1$, and for some initial conditions OGDA diverges.

In the sequel of the proof of Theorem 3.1, we always assume that $\eta < \frac{1}{\sqrt{3\mu_{\max}}}$. Consider μ in $\mathcal{S}(A)$. Since $A^T A$ and AA^T are positive semi-definite, μ is a non-negative real

number. If $\mu \neq 0$, $S(\mu) \subset \{\lambda \in \mathbb{C} : |\lambda| < 1\}$ by Lemma 3.9. If $\mu = 0$, then $S(\mu) = \{0, 1\}$. By Lemma 3.10, we have:

$$\mathrm{Sp}(\Lambda) \subset \{1\} \cup \{\lambda \in \mathbb{C} : |\lambda| < 1\} .$$

Let us define $\lambda(\eta) = \max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda) \backslash \{1\}\} < 1$. One can check that $\lambda(\eta) = \max\{\lambda_*(\eta), \lambda_{**}(\eta)\}$ with $\lambda_*(\eta) = \mathbf{1}_{\mu_{\min} \leqslant \frac{1}{4\eta^2}} \sqrt{\frac{1}{2}(1 + \sqrt{1 - 4\eta^2 \mu_{\min}})}$ and $\lambda_{**}(\eta) = \mathbf{1}_{\mu_{\max} \geqslant \frac{1}{4\eta^2}} \sqrt{2\eta^2 \mu_{\max} + \eta\sqrt{\mu_{\max}}\sqrt{4\eta^2 \mu_{\max} - 1}}$. Notice that if $\eta \leqslant \frac{1}{2\sqrt{\mu_{\max}}}$, $\lambda(\eta) = \lambda_*(\eta)$.

We now describe the eigenspaces of Λ. For each λ in $\mathbb{C}$, we write $E_\lambda = \{Z \in \mathbb{C}^{(n+m+n+m)} : \Lambda Z = \lambda Z\}$. The proof of Lemma 3.10 easily gives:

Lemma 3.12. *Let λ be in $\mathrm{Sp}(\Lambda)$, and $\mu \geqslant 0$ in $\mathcal{S}(A)$ uniquely defined by $\lambda^2(1-\lambda)^2 + \mu\eta^2(1-2\lambda)^2 = 0$.*

If $\lambda = 0$, then $\mu = 0 \in \mathcal{S}(A)$ and $E_0 = \{(x, y, x', y') \in \mathbb{C}^{(n+m+n+m)} : x = 0, y = 0, Ay' = 0, A^T x' = 0\}$, so $\dim(E_0) = \dim(\mathrm{Ker}(A)) + \dim(\mathrm{Ker}(A^T))$.

If $\lambda = 1$, then $\mu = 0 \in \mathcal{S}(A)$ and $E_1 = \{(x, y, x', y') \in \mathbb{C}^{(n+m+n+m)} : x = x', y = y', Ay' = 0, A^T x' = 0\}$, so $\dim(E_1) = \dim(\mathrm{Ker}(A)) + \dim(\mathrm{Ker}(A^T))$.

If $\lambda \notin \{0, 1\}$, then $\mu > 0$ and

$$E_\lambda = \left\{ (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)} : x = \lambda x', y = \lambda y', A^T A y' = \mu y', x' = \frac{(1-2\lambda)\eta}{\lambda(1-\lambda)} A y' \right\} ,$$

so $\dim(E_\lambda) = \dim(\mathrm{Ker}(A^T A - \mu I))$.

We can now sum the dimension of the eigenspaces.

Lemma 3.13.

$$\sum_{\lambda \in Sp(\Lambda)} \dim(E_\lambda) = 2(n+m) - 2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I)) .$$

Proof. Recall that $\mathrm{rank}(A^T A) = \mathrm{rank}(A) = \mathrm{rank}(A^T)$ and, $A^T A$ being positive semi-definite, $\mathrm{rank}(A^T A) = \sum_{\mu > 0} \dim(\mathrm{Ker}(A^T A - \mu I))$. Recall also from Lemma 3.9 that $S(\mu)$ has 2 elements if $\mu = \frac{1}{4\eta^2}$.

$$\begin{aligned}
\sum_{\lambda \in Sp(\Lambda)} \dim(E_\lambda) &= \dim(E_0) + \dim(E_1) + \sum_{\lambda \in Sp(\Lambda), \lambda \notin \{0,1\}} \dim(E_\lambda) \\
&= 2\dim(\mathrm{Ker}(A)) + 2\dim(\mathrm{Ker}(A^T)) + 2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I)) \\
&\quad + \sum_{\mu \in Sp(A^T A), \mu \neq 0, \frac{1}{4\eta^2}} 4\dim(\mathrm{Ker}(A^T A - \mu I)) \\
&= 2\dim(\mathrm{Ker}(A)) + 2\dim(\mathrm{Ker}(A^T)) + 4\mathrm{rank}(A^T A) - 2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I)) \\
&= 2(\mathrm{rank}(A) + \dim(\mathrm{Ker}(A))) + 2(\mathrm{rank}(A^T) + \dim(\mathrm{Ker}(A^T))) - 2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I)) \\
&= 2(n+m) - 2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I)) .
\end{aligned}$$

□

We can already notice that $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$ is equivalent to $\sum_{\lambda \in Sp(\Lambda)} \dim(E_\lambda) = 2(n+m)$, i.e. is equivalent to the property that Λ is diagonalizable in $\mathbb{C}$. The next lemma will be used to characterize the limit of OGDA.

Lemma 3.14. *If y is an eigenvector of A^TA associated to a non-zero eigenvalue, then y is orthogonal to* $\mathrm{Ker}(A)$. *Similarly if x is an eigenvector of AA^T associated to a non-zero eigenvalue, then x is orthogonal to* $\mathrm{Ker}(A^T)$.

Proof. Let $\widehat{y}$ be a vector in the kernel of A, and let $\mu \neq 0$ be the eigenvalue of A^TA associated with y. Then:

$$\mu\langle y, \widehat{y}\rangle = \langle \mu y, \widehat{y}\rangle = \langle A^TAy, \widehat{y}\rangle = \langle Ay, A\widehat{y}\rangle = \langle Ay, 0\rangle = 0\,.$$

Hence, $\langle y, \widehat{y}\rangle = 0$ for any $\widehat{y}$ in $\mathrm{Ker}(A)$. The proof for eigenvectors of AA^T is similar. □

3.B.2 Convergence to a Nash equilibrium if $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$

In this section, we assume that $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$, with $0 < \eta < \frac{1}{\sqrt{3\mu_{\max}}}$. Here $\mathrm{Ker}(A^TA - \frac{1}{4\eta^2}I)$ is empty, so $\sum_{\lambda \in Sp(\Lambda)} \dim(E_\lambda) = 2(n+m)$ and Λ is diagonalizable. Since $\max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda)\backslash\{1\}\} < 1$, the matrix Λ^t converges as $t \to \infty$, and the convergence is exponential with ratio $\lambda(\eta) = \max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda)\backslash\{1\}\}$.

Consider an initial condition $Z_0 = (x_0, y_0, x_{-1}, y_{-1})$. Z_0 can be uniquely written :

$$Z_0 = \sum_{\lambda \in \mathrm{Sp}(\Lambda)} z_\lambda, \text{ with } z_\lambda \in E_\lambda \text{ for all } \lambda\,.$$

For all $t \geqslant 0$, $Z_t = (x_t, y_t, x_{t-1}, y_{t-1}) = \Lambda^t Z_0 = \sum_{\lambda \in \mathrm{Sp}(\Lambda)} \lambda^t z_\lambda$ and $Z_t \xrightarrow[t\to\infty]{} Z_\infty := z_1$. We have obtained the convergence of Z_t to the projection of Z_0 onto $E_1 = \mathrm{Ker}(\Lambda - I)$ along $\bigoplus_{\lambda \neq 1} \mathrm{Ker}(\Lambda - \lambda I)$. Since $Z_\infty \in E_1$, $(x_t)_t$ converges to a limit x_∞ in $\mathrm{Ker}(A^T)$ and $(y_t)_t$ converges to a limit y_∞ in $\mathrm{Ker}(A)$. Moreover there exists a constant C', only depending on A and η through the matrix Λ, such that for all $t \geqslant 0$, $\|\Lambda^t - \Lambda^\infty\| \leqslant C'\lambda(\eta)^t$. And for all $t \geqslant 0$, $\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C'\lambda(\eta)^t$.

Now, let us write z_λ as $(\widehat{x}_\lambda, \widehat{y}_\lambda, \widehat{x}'_\lambda, \widehat{y}'_\lambda) \in \mathbb{C}^{(n+m+n+m)}$ for each λ in $\mathcal{S}(A)$. We have $y_0 = \sum_{\lambda \in \mathrm{Sp}(\Lambda)} \widehat{y}_\lambda$, and $(y_t)_t$ converges to $y_\infty = \widehat{y}_1 \in \mathrm{Ker}(A)$. $\widehat{y}_0 = 0$ and $\sum_{\lambda \neq \{0,1\}} \widehat{y}_\lambda$ is orthogonal to $\mathrm{Ker}(A)$ thanks to Lemma 3.14, so we obtain that $\widehat{y}_1 = y_\infty$ is the orthogonal projection of y_0 onto $\mathrm{Ker}(A)$. In the same way, one can prove that x_∞ is the orthogonal projection of x_0 onto $\mathrm{Ker}(A^T)$.

3.B.3 Convergence to a Nash equilibrium if $\frac{1}{4\eta^2} \in \mathcal{S}(A)$

In this section, we assume that $\frac{1}{4\eta^2} \in \mathcal{S}(A)$, with $\frac{1}{2\sqrt{\mu_{\max}}} \leqslant \eta < \frac{1}{\sqrt{3\mu_{\max}}}$.

Consider $\mu = \frac{1}{4\eta^2} > 0$. By assumption, μ is an eigenvalue of AA^T and A^TA, and $S(\mu) = \{\lambda_1, \lambda_2\}$ with $\lambda_1 = \frac{1}{2}(1+i)$ and $\lambda_2 = \frac{1}{2}(1-i)$. We know that $\dim(\mathrm{Ker}(\Lambda - \lambda_1 I)) = \dim(\mathrm{Ker}(\Lambda - \lambda_2 I)) = \dim(\mathrm{Ker}(A^TA - \mu I))$, and $\sum_{\lambda \in Sp(\Lambda)} \dim(E_\lambda) = 2(n+m) - 2\dim(\mathrm{Ker}(A^TA - \frac{1}{4\eta^2}I)) < 2(n+m)$, so Λ is not diagonalizable in $\mathbb{C}$.

If $0 \notin \mathcal{S}(A)$, then by Gelfand's theorem OGDA converges to 0, and $\mathrm{Ker}(A) = \{0\}$. To fix ideas we now assume that $0 \in \mathcal{S}(A)$.

The eigenvalues of Λ are 1 and eigenvalues with modulus < 1.

Lemma 3.15. *The algebraic multiplicity of $\lambda = 1$ in the characteristic polynomial of Λ is $\dim E_1$.*

Proof of Lemma 3.15 Let $Z = (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)}$ be such that $(\Lambda - I)^2 Z = 0$. Then $(\Lambda - I)Z \in E_1$, i.e. $2\eta Ay - \eta Ay' = x - x'$, $-2\eta A^T x + \eta A^T x' = y - y'$, $A^T(x - x') = 0$ and $A(y - y') = 0$. Then $Ay = Ay', A^T x = A^T x', \eta Ay' = x - x'$ and $-\eta A^T x' = y - y'$. It implies $A^T Ay' = 0$, which is equivalent to $Ay' = 0$. Similarly $AA^T x' = 0$, so $A^T x' = 0$ and finally $Z \in E_1$. We have shown that $\mathrm{Ker}(\Lambda - I)^2 = \mathrm{Ker}(\Lambda - I)$, so for each $l \geqslant 2$, $\mathrm{Ker}(\Lambda - I)^l = \mathrm{Ker}(\Lambda - I)$, and the algebraic multiplicity of $\lambda = 1$ is $\dim E_1$. □

Looking at the Jordan normal form of Λ, we can conclude that Λ^t converges to a projection matrix Λ^∞. For any initial condition $Z_0 = (x_0, y_0, x_{-1}, y_{-1})$, Z_t converges to an element of E_1, so $(x_t)_t$ converges to a limit x_∞ in $\mathrm{Ker}(A^T)$ and $(y_t)_t$ converges to a limit y_∞ in $\mathrm{Ker}(A)$. Moreover for every $\lambda > \lambda(\eta)$ there exists a constant C' such that for all $t \geqslant 0$, $\|\Lambda^t - \Lambda^\infty\| \leqslant C'\lambda^t$. This implies[3] $\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C'\lambda^t$ for all t. To conclude, it remains to show that x_∞, resp. y_∞, is the orthogonal projection of x_0, resp. y_0, onto $\mathrm{Ker}(A^T)$.

Lemma 3.16.

$$\dim(\mathrm{Ker}(\Lambda - \lambda_1 I)^2) = \dim(\mathrm{Ker}(\Lambda - \lambda_2 I)^2) = 2\dim(\mathrm{Ker}(A^T A - \mu I)) .$$

Moreover, if $(x, y, x', y') \in (\mathrm{Ker}(\Lambda - \lambda_1 I))^2$, then x and x' are in $(\mathrm{Ker}(A^T))^\perp$ and y and y' are in $(\mathrm{Ker}(A))^\perp$.

Proof of Lemma 3.16 We write the proof for λ_1 only. Consider $Z = (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)}$. Computations show that $Z \in (\mathrm{Ker}(\Lambda - \lambda_1 I))^2$ if and only if:

$$\begin{cases} 2\eta A(2y - y') &= i(2x - x') \\ -2\eta A^T(2x - x') &= i(2y - y') \\ \eta A(y - \lambda_1 y') &= \frac{i}{2}(x - \lambda_1 x') \\ \eta A^T(x - \lambda_1 x') &= -\frac{i}{2}(y - \lambda_1 y'). \end{cases} .$$

This implies $AA^T(2x - x') = \mu(2x - x')$ and $AA^T(x - \lambda_1 x') = \mu(x - \lambda_1 x')$.

Conversely, choose independently any 2 vectors x_1^* and x_2^* in $\mathrm{Ker}(AA^T - \mu I)$, and define uniquely x, x', y and y' by:

$$\begin{cases} 2x - x' &= x_1^* \\ x - \lambda_1 x' &= x_2^* \\ i(2y - y') &= -2\eta A^T(x_1^*) \\ -\frac{i}{2}(y - \lambda_1 y') &= \eta A^T(x_2^*). \end{cases} .$$

[3] This also holds if $0 \notin \mathcal{S}(A)$.

Then one can check that (x, y, x', y') belongs to $\mathrm{Ker}(\Lambda - \lambda_1 I)^2$.

$\mathrm{Ker}(A^T A - \mu I) \subset \mathrm{Ker}(A)^\perp$ by Lemma A.6, so x_1^* and x_2^* belong to $\mathrm{Ker}(A)^\perp$. So x and x' also belong to the vector subspace $\mathrm{Ker}(A^T)^\perp$, and similarly y and y' belong to $\mathrm{Ker}(A)^\perp$.

The map $(x_1^*, x_2^*) \mapsto (x, y, x', y')$ is linear one-to-one onto $\mathrm{Ker}(AA^T - \mu I) \times \mathrm{Ker}(A^T A - \mu I)$, so $\dim(\mathrm{Ker}(\Lambda - \lambda_1 I)^2) = 2\dim(\mathrm{Ker}(A^T A - \mu I))$. □

Looking at the dimensions, we obtain that the algebraic dimension of both eigenvalues λ_1 and λ_2 of Λ is $2\dim(\mathrm{Ker}(A^T A - \frac{1}{4\eta^2} I))$, and for $i = 1, 2$, the characteristic subspace of λ_i is $(\mathrm{Ker}(\Lambda - \lambda_i I)^2)$ for $i = 1, 2$. Using the spectral decomposition of Λ, the initial vector Z_0 can now be uniquely written:

$$Z_0 = \sum_{\lambda \in \mathrm{Sp}(\Lambda)} z_\lambda \,,$$

with $z_\lambda \in E_\lambda$ for $\lambda \notin \{\lambda_1, \lambda_2\}$ and $z_{\lambda_i} \in \mathrm{Ker}(\Lambda - \lambda_i I)^2$ for $i = 1, 2$.

$Z_t = \Lambda^t Z_0 \xrightarrow[t \to \infty]{} Z_\infty := z_1$, so $(x_t)_t$ converges to the first component $\widetilde{x}_1 \in \mathrm{Ker}(A^T)$ of z_1. By Lemmas 3.14 and 3.16, $x_0 - \widetilde{x}_1$ belong to $(\mathrm{Ker}(A^T))^\perp$, so $\widetilde{x}_1$ is the orthogonal projection of x_0 onto $\mathrm{Ker}(A^T)$. Similarly, $(y_t)_t$ converges to the orthogonal projection of y_0 onto $\mathrm{Ker}(A)$.

This concludes the proofs of parts 1) and 3) of Theorem 3.1. □

3.B.4 Speed of convergence when $0 < \eta < \frac{1}{\sqrt{3\mu_{\max}}}$ and $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$

Here, we prove part 2) of Theorem 3.1. Recall from Subsection 3.B.2 that for all $t \geqslant 1$,

$$Z_t - Z_\infty = \sum_{\lambda \in Sp(\Lambda), \lambda \neq 0,1} \lambda^t z_\lambda = \sum_{\mu \in \mathcal{S}(A)), \mu > 0} \sum_{l=1}^{4} \lambda_{l(\mu)}^t z_{\lambda_l(\mu)}.$$

The matrix Λ is not diagonalizable in an orthogonal basis, and the vectors (z_λ) are not orthogonal in general. We will show however that the situation is close to it.

From Lemma 3.12, we have for $\mu > 0$ in $\mathcal{S}(A)$, if $l = 1, 2$, $E_{\lambda_l(\mu)} =$

$$\left\{ (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)} : x = \lambda_l(\mu) x', y = \lambda_l(\mu) y', A^T A y' = \mu y', x' = \frac{-i}{\sqrt{\mu}} A y' \right\},$$

and if $l = 3, 4$, $E_{\lambda_l(\mu)} =$

$$\left\{ (x, y, x', y') \in \mathbb{C}^{(n+m+n+m)} : x = \lambda_l(\mu) x', y = \lambda_l(\mu) y', A^T A y' = \mu y', x' = \frac{i}{\sqrt{\mu}} A y' \right\}.$$

For the case where $\mu = 0$, we have $E_0 = \left\{ (0, 0, x', y') \in \mathbb{C}^{(n+m+n+m)} : A y' = 0, A^T x' = 0 \right\}$ and and $E_1 = \left\{ (x', y', x', y') \in \mathbb{C}^{(n+m+n+m)} : A y' = 0, A^T x' = 0 \right\}$.

Lemma 3.17.

a) For $\mu_1 \neq \mu_2$ in $\mathcal{S}(A)$, the vector subspaces $\bigoplus_{l=1,2,3,4} E_{\lambda_l(\mu_1)}$ and $\bigoplus_{l=1,2,3,4} E_{\lambda_l(\mu_2)}$ are orthogonal.

b) For μ in $\mathcal{S}(A)\backslash\{0\}$, $l \in \{1,2\}$ and $l' \in \{3,4\}$, the vector subspaces $E_{\lambda_l(\mu)}$ and $E_{\lambda_{l'}(\mu)}$ are orthogonal.

Proof. a) Consider $z_1 = (x_1, y_1, x_1', y_1')$ in $E_{\lambda_l(\mu_1)}$ and $z_2 = (x_2, y_2, x_2', y_2')$ in $E_{\lambda_{l'}(\mu_2)}$. We have both:

$$\langle Ay_1', Ay_2' \rangle \;=\; \langle y_1', A^T A y_2' \rangle \;=\; \langle y_1', \mu_2 y_2' \rangle \;=\; \mu_2 \langle y_1', y_2' \rangle \,,$$

$$\langle Ay_1', Ay_2' \rangle \;=\; \langle A^T A y_1', y_2' \rangle \;=\; \langle \mu_1 y_1', y_2' \rangle \;=\; \mu_1 \langle y_1', y_2' \rangle \,.$$

Since $\mu_1 \neq \mu_2$, $\langle y_1', y_2' \rangle = 0$. Then in all cases $\langle y_1, y_2 \rangle = \langle x_1', x_2' \rangle = \langle x_1, x_2 \rangle = 0$, and finally $\langle z_1, z_2 \rangle = 0$.

b) Consider $z_1 = (x_1, y_1, x_1', y_1')$ in $E_{\lambda_l(\mu)}$ and $z_2 = (x_2, y_2, x_2', y_2')$ in $E_{\lambda_{l'}(\mu)}$.

$$\begin{aligned} \langle x_1', x_2' \rangle \;&= \langle \tfrac{-i}{\sqrt{\mu}} A y_1', \tfrac{i}{\sqrt{\mu}} A y_2' \rangle \;&= -\frac{1}{\mu} \langle Ay_1', Ay_2' \rangle \\ &= -\tfrac{1}{\mu} \langle y_1', A^T A y_2' \rangle \;&= -\frac{1}{\mu} \langle y_1', \mu y_2' \rangle = -\langle y_1', y_2' \rangle \,. \end{aligned}$$

Then $\langle x_1, x_2 \rangle = -\langle y_1, y_2 \rangle$, and $\langle z_1, z_2 \rangle = \langle x_1, x_2 \rangle + \langle y_1, y_2 \rangle + \langle x_1', x_2' \rangle + \langle y_1', y_2' \rangle = 0$. □

Remark 3.18. A consequence of the previous lemma is that for $t \geqslant 1$:

$$\|Z_t - Z_\infty\|^2 = \sum_{\mu \in \mathcal{S}(A), \mu > 0} \left(\|\lambda_1(\mu)^t z_{\lambda_1(\mu)} + \lambda_2(\mu)^t z_{\lambda_2(\mu)}\|^2 + \|\lambda_3(\mu)^t z_{\lambda_3(\mu)} + \lambda_4(\mu)^t z_{\lambda_4(\mu)}\|^2 \right) .$$

We now fix $\mu > 0$ in $\mathcal{S}(A)$ and simply write λ_l for $\lambda_l(\mu)$ and z_l for z_{λ_l}.

We need to handle the terms $\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2$ and $\|\lambda_3^t z_3 + \lambda_4^t z_4\|^2$. Consider the first term (the second one will be treated similarly), we want to write $\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2 \leqslant C|\lambda_1|^{2t} \|z_1 + z_2\|^2$, where C is a constant independent of the dimension. This would be easy if z_1 and z_2 were orthogonal, and $C = 1$ would do. This would be impossible if $z_1 + z_2 = 0$ with $z_1 \neq 0$, but this can not happen since z_1 and z_2 belong to different eigenspaces. The key idea is that the "angle" between the subspaces E_{λ_1} and E_{λ_2} can not be too low and is, informally speaking, close to $\pi/4$: think of λ_1 close to 1, and of λ_2 close to 0 so that for some vector subspace F, E_{λ_1} can be seen as the set of vectors $\{(\alpha, \alpha) : \alpha \in F\}$ whereas E_{λ_2} can be seen as the set of vectors $\{(0, \alpha) : \alpha \in F\}$. If as a thought experiment we imagine F being the real line, we would have the horizontal axis and the 45-degree line in the Euclidean plane.

Notations 3.19.

$$C(\eta, \mu) = \begin{cases} \sqrt{\frac{2}{1 - \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}}} & \text{if} \quad \eta\sqrt{\mu} < 1/2 \,, \\ \sqrt{\frac{2}{1 - \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}}} & \text{if} \quad \eta\sqrt{\mu} > 1/2 \,. \end{cases}$$

Lemma 3.20. *z_l being in $E_{\lambda_l(\mu)}$ for each $l = 1,2,3,4$, we have:*
a) if $\eta\sqrt{\mu} < 1/2$,

$$|\langle z_1, z_2\rangle| \leqslant \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}\,\|z_1\|\|z_2\| \text{ and } |\langle z_3, z_4\rangle| \leqslant \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}\,\|z_3\|\|z_4\|\,,$$

and if $\eta\sqrt{\mu} > 1/2$,

$$|\langle z_1, z_2\rangle| \leqslant \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}\,\|z_1\|\|z_2\| \text{ and } |\langle z_3, z_4\rangle| \leqslant \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}\,\|z_3\|\|z_4\|\,,$$

b) $\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2 \leqslant C(\eta,\mu)^2|\lambda_1|^{2t}\|z_1+z_2\|^2$ and $|\lambda_3^t z_1 + \lambda_4^t z_2\|^2 \leqslant C(\eta,\mu)^2|\lambda_1|^{2t}\|z_3+z_4\|^2$.
Proof. a) Write $z_1 = (x_1, y_1, x_1', y_1')$ and $z_2 = (x_2, y_2, x_2', y_2')$.

$$\begin{aligned}\langle x_1', x_2'\rangle &= \langle \tfrac{-i}{\sqrt{\mu}}Ay_1', \tfrac{-i}{\sqrt{\mu}}Ay_2'\rangle &&= \frac{1}{\mu}\langle Ay_1', Ay_2'\rangle \\ &= \tfrac{1}{\mu}\langle y_1', A^TAy_2'\rangle &&= \frac{1}{\mu}\langle y_1', \mu y_2'\rangle = \langle y_1', y_2'\rangle\,,\end{aligned}$$

and $\langle x_1, x_2\rangle = \lambda_1\overline{\lambda}_2\langle x_1', x_2'\rangle = \lambda_1\overline{\lambda}_2\langle y_1', y_2'\rangle = \langle y_1, y_2\rangle$. So

$$\langle z_1, z_2\rangle = 2(1+\lambda_1\overline{\lambda}_2)\langle y_1', y_2'\rangle\,,$$

and similarly one can show that $\|z_1\|^2 = 2(1+|\lambda_1|^2)\|y_1'\|^2$ and $\|z_2\|^2 = 2(1+|\lambda_2|^2)\|y_2'\|^2$. We obtain:

$$\langle z_1, z_2\rangle = \frac{(1+\lambda_1\overline{\lambda}_2)}{\sqrt{(1+|\lambda_1|^2)(1+|\lambda_2|^2)}}\;\frac{\langle y_1', y_2'\rangle}{\|y_1'\|\|y_2'\|}\;\|z_1\|\|z_2\|\,.$$

Computations show that:
If $\eta\sqrt{\mu} < 1/2$, then $\sqrt{(1+|\lambda_1|^2)(1+|\lambda_2|^2)} = \sqrt{2+\eta^2\mu}$, and $|1+\lambda_1\overline{\lambda}_2| = \sqrt{1+5\eta^2\mu}$. Since $\frac{|\langle y_1', y_2'\rangle|}{\|y_1'\|\|y_2'\|} \leqslant 1$ by Cauchy-Schwartz inequality, we obtain the upper bound for $|\langle z_1, z_2\rangle|$. The proof is similar for $|\langle z_3, z_4\rangle|$.

If $\eta\sqrt{\mu} > 1/2$, then $\sqrt{(1+|\lambda_1|^2)(1+|\lambda_2|^2)} = \sqrt{1+5\eta^2\mu}$, and $|1+\lambda_1\overline{\lambda}_2| = \sqrt{2+\eta^2\mu}$. We obtain the upper bound for $|\langle z_1, z_2\rangle|$, and the proof is similar for $|\langle z_3, z_4\rangle|$.

b) On the one-hand,

$$\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2 \leqslant 2\left(\|\lambda_1^t z_1\|^2 + \|\lambda_2^t z_2\|^2\right) \leqslant 2|\lambda_1|^{2t}\left(\|z_1\|^2 + \|z_2\|^2\right)\,.$$

On the other hand, if $\eta\sqrt{\mu} < 1/2$,

$$\begin{aligned}\|z_1+z_2\|^2 &= \|z_1\|^2 + \|z_2\|^2 + 2\,\mathcal{R}e(\langle z_1, z_2\rangle), \\ &\geqslant \|z_1\|^2 + \|z_2\|^2 - 2\sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}\,\|z_1\|\,\|z_2\|\,, \\ &\geqslant \|z_1\|^2 + \|z_2\|^2 - \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}(\|z_1\|^2 + \|z_2\|^2), \\ &\geqslant (\|z_1\|^2 + \|z_2\|^2)\left(1 - \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}}\right)\,.\end{aligned}$$

Notice that $\eta^2\mu < 1/4$ implies $1 - \sqrt{\frac{1+5\eta^2\mu}{2+\eta^2\mu}} > 0$. We obtain:

$$\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2 \leqslant C(\eta,\mu)^2 |\lambda_1|^{2t} \|z_1 + z_2\|^2 \,.$$

And if $\eta\sqrt{\mu} > 1/2$,

$$\begin{aligned}
\|z_1 + z_2\|^2 &= \|z_1\|^2 + \|z_2\|^2 + 2\,\mathcal{R}e(\langle z_1, z_2\rangle), \\
&\geqslant \|z_1\|^2 + \|z_2\|^2 - 2\sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}\,\|z_1\|\,\|z_2\|\,, \\
&\geqslant \|z_1\|^2 + \|z_2\|^2 - \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}(\|z_1\|^2 + \|z_2\|^2), \\
&\geqslant (\|z_1\|^2 + \|z_2\|^2)\left(1 - \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}}\right) \,.
\end{aligned}$$

Notice that $\eta^2\mu > 1/4$ implies $1 - \sqrt{\frac{2+\eta^2\mu}{1+5\eta^2\mu}} > 0$. We obtain:

$$\|\lambda_1^t z_1 + \lambda_2^t z_2\|^2 \leqslant C(\eta,\mu)^2 |\lambda_1|^{2t} \|z_1 + z_2\|^2 \,.$$

The remaining inequalities are proved similarly. □

Notations 3.21. Let us denote:

$$C_*(\eta) = \mathbf{1}_{\mu_{\min} < \frac{1}{4\eta^2}} \sqrt{\frac{2}{1 - \sqrt{\frac{1+5\eta^2\mu_*}{2+\eta^2\mu_*}}}}, \text{ with } \mu_* = \max\{\mu \in \mathcal{S}(A) : \eta\sqrt{\mu} < 1/2\} \,,$$

$$C_{**}(\eta) = \mathbf{1}_{\mu_{\max} \geqslant \frac{1}{4\eta^2}} \sqrt{\frac{2}{1 - \sqrt{\frac{2+\eta^2\mu_{**}}{1+5\eta^2\mu_{**}}}}}, \text{ with } \mu_{**} = \min\{\mu \in \mathcal{S}(A) : \eta\sqrt{\mu} > 1/2\} \,,$$

$$\text{and } C(\eta) = \max\{C_*(\eta), C_{**}(\eta)\} \,.$$

We can now bound $\|Z_t - Z_\infty\|^2$.

$$\begin{aligned}
\|Z_t - Z_\infty\|^2 &= \sum_{\mu \in \mathcal{S}(A), \mu > 0} \left(\|\lambda_1(\mu)^t z_{\lambda_1(\mu)} + \lambda_2(\mu)^t z_{\lambda_2(\mu)}\|^2 + \|\lambda_3(\mu)^t z_{\lambda_3(\mu)} + \lambda_4(\mu)^t z_{\lambda_4(\mu)}\|^2 \right) \,, \\
&\leqslant \sum_{\mu \in \mathcal{S}(A), \mu > 0} C(\eta,\mu)^2 |\lambda_1(\mu)|^{2t} \left(\|z_{\lambda_1(\mu)} + z_{\lambda_2(\mu)}\|^2 + \|z_{\lambda_3(\mu)} + z_{\lambda_4(\mu)}\|^2 \right), \\
&\leqslant C(\eta)^2 \lambda(\eta)^{2t} \|Z_0\|^2 \,.
\end{aligned}$$

And we get:

$$\|Z_t - Z_\infty\| \leqslant C(\eta)\lambda(\eta)^t \|Z_0\|. \tag{3.3}$$

We can now conclude part 2) of Theorem 3.1. Take any Nash equilibrium $(x_\star, y_\star)$ in $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$, and define for each $t \geqslant -1$:

$$x'_t = x_t - x_\star \text{ and } y'_t = y_t - y_\star.$$

The sequence (x'_t, y'_t) is induced by OGDA starting at $(x'_0, y'_0, x'_{-1}, y'_{-1})$. $(x'_t, y'_t)_t$ converges to the limit $(x'_\infty, y'_\infty) = (x_\infty, y_\infty) - (x_\star, y_\star)$, so $\|(x'_\infty, y'_\infty) - (x'_t, y'_t)\| = \|(x_\infty, y_\infty) - (x_t, y_t)\|$. Using inequality (3.3), we obtain:

$$\|(x_\infty, y_\infty) - (x_t, y_t)\| \leqslant C(\eta)\lambda(\eta)^t \|Z'_0\| \,,$$

with $\|Z'_0\| = \|(x_0 - x_\star, y_0 - y_\star, x'_{-1} - x_\star, y'_{-1} - y_\star)\|$. Letting $(x_\star, y_\star)$ vary in $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$ concludes the proof of part 2) of Theorem 3.1: For all $t \geqslant 0$,

$$\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C(\eta)\ D\ \lambda(\eta)^t \,,$$

with $C(\eta) = \sqrt{\frac{2}{1 - \sqrt{\frac{1+5\eta^2\mu_{\max}}{2+\eta^2\mu_{\max}}}}}$, D is the distance from the initial condition to the set $\mathrm{Ker}(\Lambda - I)$, and $\lambda(\eta) = \sqrt{\frac{1}{2}(1 + \sqrt{1 - 4\eta^2\mu_{\min}})}$. □

3.B.5 Convergence with the linear terms

Now that we proved the convergence, with the wanted speed in the case where $b = 0$, $c = 0$ and $d = 0$, we go back to the study of the general settings where b is a payoff vector in $\mathbb{R}^n$, c a payoff vector in $\mathbb{R}^m$ and d a real number.

Proof. Notice that the fixed points of OGDA are still exactly the Nash equilibria of the game:

$$\left\{ \begin{array}{rcl} x_\star & = & x_\star + 2\eta(Ay_\star + b) - \eta(Ay_\star + b) \\ y_\star & = & y_\star - 2\eta(A^T x_\star + c) + \eta(A^T x_\star + c) \end{array} \right. \iff \left\{ \begin{array}{rl} Ay_\star + b & = 0 \\ A^T x_\star + c & = 0 \end{array} \right. .$$

Thus, saying that the set of Nash equilibria is empty is equivalent to saying that OGDA has no fixed point: if there is no Nash equilibria, OGDA diverge.

In the sequel we assume that the set of Nash equilibria is not empty, and fix a Nash equilibrium $(x_\star, y_\star)$. Define for each $t \geqslant -1$:

$$x'_t = x_t - x_\star \text{ and } y'_t = y_t - y_\star.$$

Easy calculations show that:

$$\forall t \geqslant 0, \left\{ \begin{array}{rcl} x'_{t+1} & = & x'_t + 2\eta A y'_t - \eta A y'_{t-1} \,, \\ y'_{t+1} & = & y'_t - 2\eta A^T x'_t + \eta A^T x'_{t-1} \,. \end{array} \right.$$

So the sequence $(x'_t, y'_t)_t$ follows the OGDA algorithm of Section 3.2 (see Algorithm 9). Thus, according to the previous work with null payoff vectors, the sequence $(x'_t, y'_t)_t$ converges to a Nash equilibrium $(x'_\infty, y'_\infty) \in \mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$. As a consequence $(x_t, y_t)_t$

converges to the limit $(x_\infty, y_\infty) = (x'_\infty + x_\star, y'_\infty + y_\star)$ which is a fixed point of OGDA, hence a Nash equilibrium of the game.

Denote by Π the orthogonal projection onto $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A)$. From our previous work, we have $(x'_\infty, y'_\infty) = \Pi(x'_0, y'_0)$, so:

$$(x_\infty, y_\infty) = (x_\star, y_\star) + \Pi(x_0, y_0) - \Pi(x_\star, y_\star) = (x_\star, y_\star) + \Pi\left((x_0, y_0) - (x_\star, y_\star)\right).$$

Thus (x_∞, y_∞) is the orthogonal projection of (x_0, y_0) onto the linear space $\mathrm{Ker}(A^T) \times \mathrm{Ker}(A) + (x^*, y^*)$, which is equal to $\{(x, y) : A^T x + c = 0, Ay + b = 0\}$.

It remains to study the convergence speed. For all t, we have $\|(x_t, y_t) - (x_\infty, y_\infty)\| = \|(x'_t, y'_t) - (x'_\infty, y'_\infty)\|$ and we obtain the same exponential speed as previously. If $\frac{1}{4\eta^2} \notin \mathcal{S}(A)$ we have for all $t \geqslant 0$,

$$\|(x_t, y_t) - (x_\infty, y_\infty)\| \leqslant C(\eta)\, D'\, \lambda(\eta)^t,$$

with D' the distance from $(x'_0, y'_0, x'_{-1}, y'_{-1})$ to the set $\{(x, y, x, y) \in \mathbb{R}^{(n+m+n+m)} : A^T = 0, Ay = 0\}$, and

Let us finally notice that $D' \leqslant \|(x'_0, y'_0, x'_{-1}, y'_{-1})\| = \|(x_0, y_0, x_{-1}, y_{-1}) - (x_\star, y_\star, x_\star, y_\star)\|$. Since this is true for all Nash equilibria $(x_\star, y_\star)$, we obtain that D' is not greater than the distance from $(x_0, y_0, x_{-1}, y_{-1})$ to the set $\{(x, y, x, y)\mathbb{R}^{(n+m+n+m)} : A^T x + c = 0, Ay + b = 0\}$. □

3.B.6 On the optimality of $\lambda(\eta)$

Let us now show that $\lambda(\eta)$ is optimal. For this, we need to show the following lemma:

Lemma 3.22. *Let $A \in \mathbb{R}^{n\times m}$ with $A \neq 0$, and $0 < \eta < \frac{1}{\sqrt{3\mu_{\max}}}$. Then there exist $Z'_0 \in \mathbb{R}^{n+m+n+m}$ and a constant $c > 0$ such that $\|Z'_t\| > c\lambda(\eta)^t$.*

Proof. If $\lambda(\eta) = \lambda_*(\eta)$, consider $\lambda = \frac{1}{2}(1 + \sqrt{1 - 4\eta^2\mu_{\min}} + 2i\eta\sqrt{\mu_{\min}})$, and Lemmas 3.9 and 3.10 will imply that λ is an eigenvalue of Λ with modulus $\lambda(\eta) < 1$. Similarly, if $\mu_{\max} \geqslant \frac{1}{4\eta^2}$ and $\lambda(\eta) = \lambda_{**}(\eta)$, consider $\lambda = \frac{1}{2}(1 + i\sqrt{4\eta^2\mu_{\max} - 1} + 2i\eta\sqrt{\mu_{\max}})$, and similarly Lemmas 3.9 and 3.10 will imply that λ is an eigenvalue of Λ with modulus $\lambda(\eta) < 1$.

In both cases, consider $Z_0 \in \mathbb{C}^{n+m+n+m} \neq 0$ such that $\Lambda Z_0 = \lambda Z_0$, we have for all t, $Z_t = \lambda^t Z_0 \xrightarrow[t\to\infty]{} 0$. So for all $t \geqslant 0$,

$$\|(x_t, y_t, x_{t-1}, y_{t-1}) - 0\| = \lambda(\eta)^t \|Z_0\|\,.$$

Now, the initialization Z'_0 that we should consider should have all of its coefficient real, while Z_0 may not be in $\mathbb{R}^{n+m+n+m}$. Let us prove the existence of a real vector Z'_0 verifying the lemma.

For λ an eigenvalue of Λ, let us denote by F_λ the kernel $\mathrm{Ker}(\Lambda - \lambda I)^{\nu(\lambda)}$ where $\nu(\lambda)$ is the algebraic multiplicity of λ. Then, $\mathbb{C}^{n+m+n+m} = \bigoplus_{\lambda\in \mathrm{Sp}(\Lambda)} F_\lambda$. Remark that if Λ is diagonalizable, $F_\lambda = E_\lambda$ for all λ. Because $\mathrm{Vect}_{\mathbb{C}}(\mathbb{R}^{n+p+n+p}) = \mathbb{C}^{n+p+n+p}$, there exist a

vector $Z'_0 \in \mathbb{R}^{n+m+n+m}$ such that its projection on $F_{\lambda(\eta)}$ along $\bigoplus_{\lambda \in \mathrm{Sp}(\Lambda)\setminus\{\lambda(\eta)\}} F_\lambda$ is non-zero. Let us call $z_{\lambda(\eta)}$ this projection. Then, because $\lambda(\eta)$ is the biggest eigenvalue strictly smaller than 1, there exits a constant $c > 0$ with

$$\left\|\Lambda^t(Z'_0 - Z'_\infty)\right\| \sim \left\|\Lambda^t z_{\lambda(\eta)}\right\| > c\lambda(\eta)^t$$

proving the wanted result.

□

Note that we used the notation $\sim$ to say that two functions are asymptotically equivalent: $f(t) \sim g(t) \iff \lim_{t\to+\infty} \frac{f(t)}{g(t)} = 1$.

3.C Proof of Theorem 3.3

Let us fix $A, B \in \mathbb{R}^{n\times m} \neq 0$, $b, e \in \mathbb{R}^n$, $c, f \in \mathbb{R}^m$ and $\eta > 0$.

Once again, we need to introduce some new notations.

Definition 3.23. We denote by $\mathcal{S}(A, B)$ the set $\mathrm{Sp}(B^TA) \cup \mathrm{Sp}(AB^T)$ (recall that B^TA and AB^T have the same non-zero eigenvalues).
Furthermore, for $\mu \in C$, define $S^*(\mu) = \{\lambda \in \mathbb{C} : \lambda^2(1-\lambda)^2 = \mu\eta^2(1-2\lambda)^2\}$.

Now, as in the previous proof, we need to define the matrix linked to the dynamical system defined by OGDA.

Definition 3.24. We define the matrix $\Lambda_{A,B}$ and the vector u by

$$\Lambda_{A,B} = \begin{pmatrix} I_n & 2\eta A & 0 & -\eta A \\ 2\eta B^T & I_m & -\eta B^T & 0 \\ I_n & 0 & 0 & 0 \\ 0 & I_m & 0 & 0 \end{pmatrix} \in \mathbb{R}^{(n+m+n+m)\times(n+m+n+m)},\ u = \begin{pmatrix} \eta b \\ \eta f \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^{(n+m+n+m)}.$$

Once more, for $t \geqslant 0$, we let Z_t be the column vector $Z_t = (x_t, y_t, x_{t-1}, y_{t-1}) \in \mathbb{R}^{(n+m+n+m)}$.

With this definition, we can now see OGDA as a dynamical system in $\mathbb{R}^{(n+p+n+p)}$, with $Z_0 \in \mathbb{R}^{(n+p+n+p)}$ and $\forall t \geqslant 0, Z_{t+1} = \Lambda_{A,B}Z_t + u$.

For $b, e = 0$ and $c, f = 0$, vector u equals 0, and thus $Z_{t+1} = \Lambda_{A,B}Z_t = \Lambda^{t+1}_{A,B}Z_0$, and the study of the dynamical system above depends only on the eigenvalues of $\Lambda_{A,B}$.
In a first part, we characterize the eigenvalues of $\Lambda_{A,B}$ in function of the element of $\mathcal{S}(A, B)$. Secondly, we study the case where all vectors are 0. Finally, we use the previous study to go back to the more general settings where b, e (respectively c, f) can be set to any vector in $\mathbb{R}^n$ (respectively $\mathbb{R}^m$).

3.C.1 Description of the eigenvalues of $\Lambda_{A,B}$

Proposition 3.25. $\mathrm{Sp}(\Lambda_{A,B}) = \bigcup_{\mu\in\mathcal{S}(A,B)} S^*(\mu)$.

Proof. For $\lambda\in\mathbb{C}$ and $Z=(x,y,x',y')\in\mathbb{C}^{(n+m+n+m)}$, we have:

$$\Lambda Z=\lambda Z \iff \begin{cases} x+2\eta Ay-\eta Ay'=\lambda x \\ y+2\eta B^Tx-\eta B^Tx'=\lambda y \\ x=\lambda x' \\ y=\lambda y' \end{cases}$$
$$\iff \begin{cases} \lambda(1-\lambda)x'+(2\lambda-1)\eta Ay'=0 \\ \lambda(1-\lambda)y'+(2\lambda-1)\eta B^Tx'=0 \\ x=\lambda x' \\ y=\lambda y' \end{cases}.$$

Assume $\Lambda Z=\lambda Z$ and $Z\neq 0$. Multiplying (from the left) the first line by $(2\lambda-1)\eta B^T$ and the second line by $(2\lambda-1)\eta A$, we get:

$$\begin{cases} \lambda^2(1-\lambda)^2y'-(2\lambda-1)^2\eta^2B^TAy'=0 \\ \lambda^2(1-\lambda)^2x'-(2\lambda-1)^2\eta^2AB^Tx'=0\,. \end{cases} \tag{3.4}$$

Since $Z\neq 0$, we have $x'\neq 0$ or $y'\neq 0$, and $\lambda\neq 1/2$. This implies that $\frac{\lambda^2(1-\lambda)^2}{(2\lambda-1)^2\eta^2}$ is an eigenvalue of B^TA or of AB^T. If we denote by μ such eigenvalue, then $\lambda\in S^*(\mu)$. This gives us the first inclusion.

Conversely, let μ be an eigenvalue of AB^T, and consider $\lambda\in S^*(\mu)$.
If $\mu\neq 0$, let $x'\neq 0$ be such that $AB^Tx'=\mu x'$ and consider y' satisfying:

$$\lambda(1-\lambda)y'=-\eta(2\lambda-1)B^Tx'.$$

Because $\mu\neq 0$, $\lambda\notin\{0,1\}$, and thus y' is uniquely defined. Setting $Z_\lambda=(\lambda x',\lambda y',x',y')$ gives us an eigenvector of $\Lambda_{A,B}$ associated with λ.
If $\mu=0$ then $AB^Tx'=0$. If $B^Tx'=0$, then setting $Z_0=(0,0,x',0)$ and $Z_1=(x',0,x',0)$ will give us an eigenvector of $\Lambda_{A,B}$ associated with 0 and 1 respectively. Otherwise, if $B^Tx'\neq 0$, then we set $Z_0=(0,0,0,B^Tx')$ and $Z_1=(0,B^Tx',0,B^Tx')$ to have an eigenvector of $\Lambda_{A,B}$ associated with 0 and 1 respectively. We have obtained that for each eigenvalue μ of AB^T, each $\lambda\in\mathcal{S}^*(\mu)$ is an eigenvalue of Λ. Similarly, this also stands for all μ an eigenvalue of B^TA.

Thus, we proved that the set of eigenvalues of $\Lambda_{A,B}$ is $\{\lambda:\lambda\in S^*(\mu),\mu\in\mathcal{S}(A,B)\}$. □

This proof provides useful expressions for the eigenspaces $E_\lambda=\mathrm{Ker}(\Lambda_{A,B}-\lambda I_{2(n+m)})$, that we will need in the proof for the next theorem. To summarize, we showed that for $\lambda\neq 0,1$,

$$E_\lambda=\left\{(x,y,x',y')\in\mathbb{C}^{n+m+n+m}: x=\lambda x', y=\lambda y', B^TAy'=\mu y', x'=\frac{(1-2\lambda)\eta Ay'}{\lambda(1-\lambda)}\right\}.$$

and that

$$E_0 = \left\{(0,0,x',y') \in \mathbb{C}^{n+m+n+m} : x' \in \mathrm{Ker}(B^T), y' \in \mathrm{Ker}(A)\right\},$$

$$E_1 = \left\{(x',y',x',y') \in \mathbb{C}^{n+m+n+m} : x' \in \mathrm{Ker}(B^T), y' \in \mathrm{Ker}(A)\right\}.$$

3.C.2 Convergence without linear terms

The first lemma follows from simple computations.

Lemma 3.26. *Assume that $\mathcal{S}(A,B) \subset \mathbb{R}$, and $\eta < \frac{1}{2\sqrt{\mu_{\max}}}$, where $\mu_{\max} = \rho(B^TA) = \rho(AB^T)$. Then, for $\mu \in \mathcal{S}(A,B)$, there are three cases:*
- If $\mu < 0$, define $\nu = i\sqrt{|\mu|}$ and $\delta = \sqrt{1-4\eta^2|\mu|} \geqslant 0$. Then $S^*(\mu)$ has exactly 4 elements, which are $\lambda_1^*(\mu) = \frac{1}{2}(1+\delta+2\eta\nu)$, $\lambda_2^*(\mu) = \frac{1}{2}(1-\delta+2\eta\nu)$, $\lambda_3^*(\mu) = \overline{\lambda_1^*}(\mu)$, $\lambda_4^*(\mu) = \overline{\lambda_2^*}(\mu)$. Their modulus are all strictly smaller than 1.
- If $\mu = 0$, then $S^*(\mu)$ possess two elements: $S^*(\mu) = \{0,1\}$.
- If $\mu > 0$, define $\nu = \sqrt{\mu}$ and $\delta = \sqrt{1+4\eta^2\nu^2}$. Then $S^*(\mu)$ has exactly 4 elements which are real, denoted: $\lambda_1^*(\mu) = \frac{1}{2}(1+2\eta\nu+\delta) > 1$, $\lambda_2^*(\mu) = \frac{1}{2}(1+2\eta\nu-\delta) \in (0,1)$, $\lambda_3^*(\mu) = \frac{1}{2}(1-2\eta\nu+\delta) \in (0,1)$ and $\lambda_4^*(\mu) = \frac{1}{2}(1-2\eta\nu-\delta) \in (-1,0)$.

We now show that under the condition of the theorem, $\Lambda_{A,B}$ is diagonalizable.

Lemma 3.27. *Let A,B be in $\mathbb{R}^{n\times m}$. Assume that B^TA is diagonalizable, that $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$ and that $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$.*
Then $\Lambda_{A,B}$ is diagonalizable.
Besides, we get that $\mathrm{Im}(B^TA) = \mathrm{Im}(B^T)$ and $\mathrm{Im}(AB^T) = \mathrm{Im}(A)$.

Proof. From the hypotheses, we get that $\dim(\mathrm{Ker}(A)) + \mathrm{rank}(B^T) = m$ and $\dim(\mathrm{Ker}(B^T)) + \mathrm{rank}(A) = n$. Using the rank-nullity theorem, we get that $\mathrm{rank}(B^T) = \mathrm{rank}(A)$.
Let us now show that $\mathrm{rank}(B^TA) = \mathrm{rank}(B^T)$.
We have that $\mathrm{Im}(B^TA) \subset \mathrm{Im}(B^T)$. Conversely, for $y \in \mathrm{Im}(B^T)$, there exists $x \in \mathbb{C}^n$ such that $y = B^Tx$. Because $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$, there exists $x', x'' \in \mathrm{Ker}(B^T) \times \mathrm{Im}(A)$ such that $x = x' + x''$. Then, $y = B^Tx = B^T(x'+x'') = B^Tx''$ because $x' \in \mathrm{Ker}(B^T)$, showing that $y \in B^T\mathrm{Im}(A) = \mathrm{Im}(B^TA)$.
This shows that $\mathrm{Im}(B^T) \subset \mathrm{Im}(B^TA)$, and thus that these two sets are equal: $\mathrm{Im}(B^TA) = \mathrm{Im}(B^T)$, and thus $\mathrm{rank}(B^TA) = \mathrm{rank}(B^T)$. In a similar way, we can also show that $\mathrm{Im}(AB^T) = \mathrm{Im}(A)$.

Because B^TA is assumed to be diagonalizable, we have:

$$\mathrm{rank}(B^TA) = \sum_{\mu \in \mathrm{Sp}(B^TA), \mu \neq 0} \dim(\mathrm{Ker}(B^TA - \mu I_p)).$$

Then, from Proposition 3.25, we get:

$$\begin{aligned}\sum_{\lambda\in Sp(\Lambda_{A,B})}\dim(E_\lambda) &= \dim(E_0)+\dim(E_1)+\sum_{\lambda\in Sp(\Lambda_{A,B}),\lambda\notin\{0,1\}}\dim(E_\lambda)\\ &= 2\dim(\mathrm{Ker}(A))+2\dim(\mathrm{Ker}(B^T))+\sum_{\mu\in Sp(B^TA),\mu\neq 0}4\dim(\mathrm{Ker}(B^TA-\mu I))\\ &= 2\dim(\mathrm{Ker}(A))+2\dim(\mathrm{Ker}(B^T))+4\mathrm{rank}(B^TA)\\ &= 2\dim(\mathrm{Ker}(A)))+2\dim(\mathrm{Ker}(B^T))+2(\mathrm{rank}(A)+\mathrm{rank}(B^T))\\ &= 2(n+m)\,.\end{aligned}$$

This shows that $\Lambda_{A,B}$ is diagonalizable. □

Proof of Theorem 3.3 without linear terms. Assume that $\mathcal{S}(A,B)\subset\mathbb{R}_{\leqslant 0}$. Then, using Proposition 3.25 and Lemma 3.26, we deduce that for $\eta<\frac{1}{2\sqrt{\mu_{\max}}}$ we have

$$\mathrm{Sp}(\Lambda_{A,B})\subset\{1\}\cup\{\lambda\in\mathbb{C}:|\lambda|<1\}\,.$$

a) Assume A and B are square invertible. Then $\mathrm{Sp}(AB^T)=\mathrm{Sp}(B^TA)\subset\{\mu\in\mathbb{R}:\mu<0\}$, and for $0<\eta<\frac{1}{2\sqrt{\mu_{\max}}}$, $\mathrm{Sp}(\Lambda_{A,B})\subset\{\lambda\in\mathbb{C}:|\lambda|<1\}$. Then $\left\|\Lambda_{A,B}^t\right\|^{\frac{1}{t}}\xrightarrow[t\to\infty]{}\rho(\Lambda_{A,B})<1$ by Gelfand's theorem, so $\left\|\Lambda_{A,B}^tZ_0\right\|$ is of order $\rho(\Lambda_{A,B})^t\|Z_0\|$. It proves that (x_t,y_t) tends to $(0,0)$ exponentially fast as t tends to infinity. The result follows since $\mathrm{Ker}(A)=\mathrm{Ker}(B^T)=\{0\}$.

b) Let $Z_0=(x_0,y_0,x_{-1},y_{-1})$ and $Z_t=(x_t,y_t,x_{t-1},y_{t-1})=\Lambda_{A,B}^tZ_0$.

From Lemma 3.27, we know that $\Lambda_{A,B}$ is diagonalizable. Since, in addition its spectrum is included in $\{1\}\cup\{\lambda\in\mathbb{C},|\lambda|<1\}$, the sequence $(\Lambda_{A,B}^tZ_0)_t$ converges exponentially fast, with speed $\max\{|\lambda|,\lambda\in\mathrm{Sp}(\Lambda_{A,B}),\lambda\neq 1\}$, to the projection of Z_0 onto the eigenspace associated to the eigenvalue 1, along the direct sum of all the other eigenspaces. Moreover, $(x_\infty,y_\infty,x_\infty,y_\infty)\in\mathrm{Ker}(\Lambda_{A,B}-I)$ implies that (x_∞,y_∞) is in $\mathrm{Ker}(B^T)\times\mathrm{Ker}(A)$. This proves that $(x_t,y_t)_t$ converges exponentially fast to a Nash equilibrium.

Since $\Lambda_{A,B}$ is diagonalizable,

$$\mathbb{C}^{(n+m+n+m)}=\bigoplus_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})}\mathrm{Ker}(\Lambda_{A,B}-\lambda I)\,.$$

Let $Z_0=(x_0,y_0,x_{-1},y_{-1})$ denote the initialization vector. We can decompose Z_0 as $\sum_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})}Z_\lambda$ where each $Z_\lambda\in E_\lambda$ is itself decomposed as $Z_\lambda=(\widehat{x}_\lambda,\widehat{y}_\lambda,\widetilde{x}_\lambda,\widetilde{y}_\lambda)$. This implies that $x_0=\sum_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})}\widehat{x}_\lambda$ and that $y_0=\sum_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})}\widehat{y}_\lambda$.
Yet, from the expression of the eigenspaces E_λ seen in the proof of Proposition 3.25, we know that $\widehat{x}_0=0$, $\widehat{y}_0=0$, $\widehat{x}_1\in\mathrm{Ker}(B^T)$ and $\widehat{y}_1\in\mathrm{Ker}(A)$. Moreover, for each $\lambda\notin\{0,1\}$, $\widehat{x}_\lambda$ is an eigenvector of AB^T associated to a non-zero eigenvalue, and similarly $\widehat{y}_\lambda$ is an eigenvector of B^TA associated to a non-zero eigenvalue. Then Lemma 3.27 gives that $\sum_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})\setminus\{0,1\}}\widehat{x}_\lambda\in\mathrm{Im}(A)$ and $\sum_{\lambda\in\mathrm{Sp}(\Lambda_{A,B})\setminus\{0,1\}}\widehat{y}_\lambda\in\mathrm{Im}(B^T)$.
Yet, from the assumptions of the theorem, $\mathrm{Ker}(B^T)$ and $\mathrm{Im}(A)$ are in direct sum, and

thus $x_\infty = \widehat{x}_1$ is the linear projection of x_0 onto $\mathrm{Ker}(B^T)$ along $\mathrm{Im}(A)$. We also know from the assumptions that $\mathrm{Ker}(A)$ and $\mathrm{Im}(B^T)$ are in direct sum. Thus $y_\infty = \widehat{y}_1$ is the linear projection of y_0 onto $\mathrm{Ker}(A)$ along $\mathrm{Im}(B^T)$.

In both cases a) and b), the convergence is exponential with rate given by $\max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda_{A,B}), \lambda \neq 1\}$ that is

$$|\lambda_1^*(\mu)| = \max\left\{ \sqrt{\frac{1}{2}(1+\sqrt{1+4\eta^2\mu})} : \mu < 0, \mu \in \mathcal{S}(A,B) \right\} .$$

□

3.C.3 Convergence with linear terms

Now that we proved the convergence, with the wanted speed in the case where $b, e = 0$ and $c, f = 0$, we go back to the study of the general settings where b and e are payoffs vectors in $\mathbb{R}^n$ and c and f are payoffs vectors in $\mathbb{R}^m$.

Proof. We proceed as in Section 3.B.5 of the proof for Theorem 3.1. Fix a Nash equilibrium $(x_\star, y_\star)$. Define for each $t \geqslant -1$, $x'_t = x_t - x_\star$ and $y'_t = y_t - y_\star$. Easy calculations show that:

$$\forall t \geqslant 0, \begin{cases} x'_{t+1} &= x'_t + 2\eta A y'_t - \eta A y'_{t-1} , \\ y'_{t+1} &= y'_t - 2\eta B^T x'_t + \eta B^T x'_{t-1} . \end{cases}$$

So the sequence $(x'_t, y'_t)_t$ follows the OGDA algorithm of Section 3.3 (Algorithm 10). Thus, according to what was done previously, when all vector payoff were set to 0, the sequence $(x'_t, y'_t)_t$ converges to the limit (x'_∞, y'_∞), where x'_∞ is the projection of x'_0 on $\mathrm{Ker}(B^T)$ along $\mathrm{Ker}(A^T)^\perp$ and y'_∞ is the projection of y'_0 on $\mathrm{Ker}(A)$ along $\mathrm{Ker}(B)^\perp$. As a consequence $(x_t, y_t)_t$ converges to the limit $(x_\infty, y_\infty) = (x'_\infty + x_\star, y'_\infty + y_\star)$ which is a fixed point of OGDA, hence a Nash equilibrium of the game. Moreover $x_\infty - x_0 = x'_\infty - x'_0 \in \mathrm{Ker}(A^T)^\perp$, so x_∞ is the projection of x_0 on $\{x \in \mathbb{R}^n : B^T x + f = 0\}$ along $\mathrm{Ker}(A^T)^\perp$. Similarly y_∞ is the projection of y_0 on $\{y \in \mathbb{R}^m : Ay + b = 0\}$ along $\mathrm{Ker}(B)^\perp$.

And part *3)* of Theorem 3.3 holds since $\|(x_t, y_t) - (x_\infty, y_\infty)\| = \|(x'_t, y'_t) - (x'_\infty, y'_\infty)\|$. □

3.D Proof of Theorem 3.4

We start with a lemma.

Lemma 3.28. *Let $A \in \mathbb{R}^{n\times m}$ and $B = -(A^\dagger)^T$. Then $\mathrm{Sp}(B^T A) \subset \mathbb{R}_{\leqslant 0}$ and for $0 < \eta < \frac{1}{2}$, $B^T A$ is diagonalizable, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$ and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$.*

Proof. We recall that $B = -(A^\dagger)^T$ Let $A = U\Sigma V^T$ be the singular value decomposition of A. Then $A^\dagger = V\Sigma^\dagger U^T$. Thus $B^T A = -A^\dagger A = V\Sigma^\dagger \Sigma V^T$ is diagonalizable with eigenvalues 0 and 1, due to the value of $\Sigma^\dagger\Sigma$.

Moreover, a property of the Moore-Penrose inverse is that $\mathrm{Ker}(A^\dagger) = \mathrm{Ker}(A^T)$ and $\mathrm{Im}(A^\dagger) = \mathrm{Im}(A)$. This proves the wanted results because we have both

$$\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathrm{Ker}(A) \oplus \mathrm{Im}(A^\dagger) = \mathrm{Ker}(A) \oplus \mathrm{Im}(A^T) = \mathbb{C}^m ,$$

and

$$\mathrm{Im}(A) \oplus \mathrm{Ker}(B^T) = \mathrm{Im}(A) \oplus \mathrm{Ker}(A^\dagger) = \mathrm{Im}(A) \oplus \mathrm{Ker}(A^T) = \mathbb{C}^n \,.$$

□

Proof of Theorem 3.4. The proof of Theorem 3.4 now follows from Theorem 3.3 and Lemma 3.28. We only need to look at the case where $\eta = 1/2$.

Then $\mathcal{S}(A,B) \subset \{0,-1\}$, and $\mathrm{Sp}(\Lambda_{A,B}) \subset \left\{0, 1, \frac{1}{2}(1+i), \frac{1}{2}(1-i)\right\}$. As in the proof of Lemma 3.15 one can show that $\mathrm{Ker}(\Lambda_{A,B} - I)^2) = \mathrm{Ker}(\Lambda_{A,B} - I) = \{(x,y,x',y') \in \mathbb{C}^{(n+m+n+m)} : x = x', y = y', A^\dagger x' = 0, Ay' = 0\}$. So the algebraic multiplicity of $\lambda = 1$ in the characteristic polynomial of Λ is $\dim E_1$, and looking at the Jordan normal form of $\Lambda_{A,B}$ we can conclude for the speed of convergence in the case where $\eta = 1/2$. □

3.E Proof of Theorem 3.5

In a first part, we prove in Lemmas 3.29 and 3.30 that $B^T A$ is diagonalizable, that $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$, that $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$, and $\mathcal{S}(A,B)$ is included in $\mathbb{R}$ both when $B = \alpha A$ for $\alpha \neq 0$, and when A and B are both square matrices in $\mathbb{R}^{n\times n}$ such that $B^T A$ is diagonalizable with n distinct non-zero real eigenvalues.

Lemma 3.29. *Assume that $\eta < \frac{1}{2\sqrt{\mu_{\max}}}$, and that $A = \alpha B$ for some real $\alpha \neq 0$. Then $\mathcal{S}(A,B) \subset \mathbb{R}$, $B^T A$ is diagonalizable, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^m$, and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$.*

Proof. Here $A = \alpha B$ with $\alpha \neq 0$. Then,

$$\mathcal{S}(A,B) = \mathrm{Sp}(\alpha A^T A) \cup \mathrm{Sp}(\alpha A A^T) = \alpha(\mathrm{Sp}(A^T A) \cup \mathrm{Sp}(A A^T)) \subset \mathbb{R} \,.$$

Moreover, $B^T A = -A^T A$ is diagonalizable, because it is symmetric. Finally, from $\mathrm{Ker}(\alpha A) = \mathrm{Ker}(A)$ and $\mathrm{Im}(\alpha A) = \mathrm{Im}(A)$, we have

$$\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathrm{Ker}(A) \oplus \mathrm{Im}(A^T) = \mathbb{C}^m \,,$$

and

$$\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathrm{Ker}(A^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n \,.$$

It follows that all the hypotheses of Theorem 3.5 are respected. □

Lemma 3.30. *Assume that $\eta < \frac{1}{2\sqrt{\mu_{\max}}}$, and that A and B are square matrices in $\mathbb{R}^{n\times n}$ such that $B^T A$ is diagonalizable with real eigenvalues. Then $\mathcal{S}(A,B) \subset \mathbb{R}$, $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^n$ and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$.*

Proof. Assume that A and B are square invertible matrices with $B^T A$ diagonalizable with real eigenvalues. Then, because $B^T A$ and AB^T are invertible, $\mathcal{S}(A,B) = \mathrm{Sp}(B^T A) \subset \mathbb{R}$. Besides, $\mathrm{Im}(A) = \mathrm{Im}(B^T) = \mathbb{C}^n$ and $\mathrm{Ker}(A) = \mathrm{Ker}(B^T) = \{0\}$. It easily follows that $\mathrm{Ker}(A) \oplus \mathrm{Im}(B^T) = \mathbb{C}^n$ and $\mathrm{Ker}(B^T) \oplus \mathrm{Im}(A) = \mathbb{C}^n$.
It follows that all the hypotheses of Theorem 3.5 are respected. □

Now that Lemmas 3.29 and 3.30 are proved, we want to prove the heart of the theorem, that is the convergence to a Nash equilibrium or to infinite payoffs for matrices respecting the hypotheses.
We recall that from Proposition 3.25 we get that: $\mathrm{Sp}(\Lambda_{A,B}) = \bigcup_{\mu\in\mathcal{S}(A,B)} S^*(\mu)$, with $S^*(\mu) = \{\lambda\in\mathbb{C} : \lambda^2(1-\lambda)^2 = \mu\eta^2(1-2\lambda)^2\}$.

Proof of Theorem 3.5. For the first part of the proof, let us assume that b, c, d, e, f and g are zero vectors. We will then use it to cope with the more general case where they can be non-zero.
We have for each t, $Z_t = \Lambda_{A,B}^t Z_0$, with Z_t the column vector $(x_t^T, y_t^T, x_{t-1}^T, y_{t-1}^T)^T$. From Lemma 3.27, we know that $\Lambda_{A,B}$ is diagonalizable, and so, Z_0 can be uniquely written as $Z_0 = \sum_{\mu\in\mathcal{S}(A,B)}\sum_{\lambda\in S^*(\mu)} z_{0,\lambda}$, with $z_{0,\lambda}\in E_\lambda$ for each λ, where E_λ is the eigenspace associated with λ.
Then for each t,

$$Z_t = \sum_{\mu\in\mathcal{S}(A,B)}\sum_{\lambda\in S^*(\mu)} \lambda^t z_{0,\lambda}\,.$$

Define $\widehat{\rho} = \max\{|\lambda| \in \mathrm{Sp}(\Lambda_{A,B}) : z_{0,\lambda}\neq 0\}$, and $\widehat{\lambda}$ an eigenvalue verifying $|\widehat{\lambda}| = \widehat{\rho}$.

If $\widehat{\rho} < 1$, then $\|Z_t\| \sim \widehat{\rho}^t$, so (x_t, y_t) converges with exponential speed to $(0,0)$, which is a Nash equilibrium of the game.

If $\widehat{\rho} = 1$, looking at Lemma 3.26, we can see that $\widehat{\lambda} = 1$ is the only possible option such that $|\widehat{\lambda}| = 1$. Then $Z_t \xrightarrow[t\to\infty]{} z_{0,1}\in E_1^*$, where there exist $(x_\infty, y_\infty)\in\mathbb{R}^{(n+m)}$ such that $z_{0,1} = (x_\infty, y_\infty, x_\infty, y_\infty)$, with $Ay_\infty = 0$ and $B^T x_\infty = 0$. Now, let $\rho_{**} = \max\{|\lambda| \in \mathrm{Sp}(\Lambda_{A,B})\backslash\{1\} : z_{0,\lambda}\neq 0\}$, then $\rho_{**} < 1$, $\|Z_t - z_{0,1}\| \sim \rho_{**}^t$ and thus $(x_t, y_t)_t$ converges exponentially fast to the Nash equilibrium (x_∞, y_∞).

Assume finally that $\widehat{\rho} > 1$, we will prove that $x_t^T A y_t \xrightarrow[t\to\infty]{} +\infty$ and $x_t^T A y_t \xrightarrow[t\to\infty]{} +\infty$.
Let $\widehat{\lambda}$ be such that $|\widehat{\lambda}| = \widehat{\rho}$. According to Lemma 3.26, $\widehat{\lambda}$ is a positive real: $\widehat{\lambda} = \widehat{\rho} > 1$, and $\widehat{\lambda}\in S^*(\mu_*)$, with $\mu_* > 0$ the largest element of $\mathcal{S}(A,B)$.
Consider $z_{0,\widehat{\lambda}} = (x, y, x', y')^T \in E_{\widehat{\lambda}}$, we have $z_{0,\widehat{\lambda}}\neq 0$, $x = \widehat{\lambda}x'$, $y = \widehat{\lambda}y'$, $B^T A y' = \mu_* y'$ and $x' = \frac{(1-2\widehat{\lambda})\eta}{\widehat{\lambda}(1-\widehat{\lambda})} A y'$. Since $\widehat{\lambda} > 1$, we not only have $\widehat{\lambda}\in S^*(\mu_*)$, but we also have $\widehat{\lambda}(1-\widehat{\lambda}) = \sqrt{\mu_*}\eta(1-2\widehat{\lambda})$.
We obtain:

$$\langle x', Ay'\rangle = \frac{1}{\sqrt{\mu_*}}\|Ay'\|^2 > 0, \text{ and thus } \langle x, Ay\rangle = \frac{\widehat{\lambda}^2}{\sqrt{\mu_*}}\|Ay'\|^2 > 0\,,$$

$$\langle B^T x', y'\rangle = \sqrt{\mu_*}\|y'\|^2 > 0, \text{ and thus } \langle B^T x, y\rangle = \widehat{\lambda}^2\sqrt{\mu_*}\|y'\|^2 > 0\,.$$

Now, $x_t \sim \widehat{\lambda}^t x$ and $y_t \sim \widehat{\lambda}^t y$ when $t\to\infty$, so the payoff $\langle x_t, Ay_t\rangle$ of player 1 is equivalent to $\widehat{\lambda}^{2t}\langle x, Ay\rangle$ and we get:

$$\langle x_t, Ay_t\rangle \sim \widehat{\lambda}^{2t}\langle x, Ay\rangle \xrightarrow[t\to\infty]{} +\infty\,.$$

Similarly,

$$\langle B^T x_t, y_t\rangle \sim \widehat{\lambda}^{2t}\langle B^T x, y\rangle \xrightarrow[t\to\infty]{} +\infty\,.$$

We now consider the case where b, c, d, e, f and g are arbitrary. We proceed as in the end of the proof of Theorem 3.3. Fix a Nash equilibrium $(x_\star, y_\star)$. Define for each $t \geqslant -1$, $x'_t = x_t - x_\star$ and $y'_t = y_t - y_\star$. Easy calculations show that:

$$\forall t \geqslant 0,\ \begin{cases} x'_{t+1} &= x'_t + 2\eta A y'_t - \eta A y'_{t-1}\,, \\ y'_{t+1} &= y'_t - 2\eta B^T x'_t + \eta B^T x'_{t-1}\,. \end{cases}$$

So the sequence $(x'_t, y'_t)_t$ follows OGDA for general-sum games (Algorithm 10). Now, according to the first part of the proof, there are two cases depending on $\widehat{\rho} = \max\{|\lambda| : \lambda \in \mathrm{Sp}(\Lambda_{A,B}), z_{0,\lambda} \neq 0\}$:

Either $\widehat{\rho} \leqslant 1$, in which case $(x'_t, y'_t)_t$ will converge to exponentially fast to a point $(x'_\infty, y'_\infty) \in \mathrm{Ker}(B^T) \times \mathrm{Ker}(A)$. As a consequence, $(x_t, y_t)_t$ converges to the limit $(x_\infty, y_\infty) = (x'_\infty + x^*, y'_\infty + y^*)$, which is a fixed point of OGDA, and thus a Nash equilibrium of the game.

Or $\widehat{\rho} > 1$, in which case $x'_t \sim \widehat{\lambda}^t x'$ and $y'_t \sim \widehat{\lambda}^t y$. From the fact that $x_t = x'_t + x^*$ and $y_t = y'_t + y^*$, we have $x_t \sim \widehat{\lambda}^t x'$ and $y_t \sim \widehat{\lambda}^t y'$. We finally have for the payoffs at stage t:

$$x_t^T A y_t + b^T x_t + c^T y_t + d \sim \widehat{\lambda}^{2t} x'^T A y' + \widehat{\lambda}^t b^T x' + \widehat{\lambda}^t c^T y' + d \sim \widehat{\lambda}^{2t} x'^T A y' \xrightarrow[t\to\infty]{} +\infty\,,$$

and

$$x_t^T B y_t + e^T x_t + f^T y_t + g \sim \widehat{\lambda}^{2t} x'^T B y' + \widehat{\lambda}^t e^T x' + \widehat{\lambda}^t f^T y' + g \sim \widehat{\lambda}^{2t} x'^T B y' \xrightarrow[t\to\infty]{} +\infty\,,$$

because, as before, $x'^T A y'$ and $x'^T B y'$ are positive. □

3.F Proof of Proposition 3.6

Proof. A and B being real matrices, if μ is an eigenvalue of $B^T A$, then $\overline{\mu}$ is also an eigenvalue of $B^T A$.
Let A and B be in $\mathbb{R}^{2\times 2}$ such that the two eigenvalues μ and $\overline{\mu}$ of $B^T A$ are complex.

Assumption 1. Assume that $S^*_\eta(\mu)$ possesses one and only one eigenvalue λ with $|\lambda| > 1$ and that λ is not a real number.

In this case, we have $\overline{\lambda} \in S^*_\eta(\overline{\mu})$, once again because $\Lambda_{A,B}$ is real, and so the conjugate of λ is one of its eigenvalue.
Let $v \in \mathbb{C}^8$ be a non-zero eigenvector associated to λ. Then $\overline{v}$ is an eigenvector associated with $\overline{\lambda}$.

Now, let us take $Z_0 = v + \overline{v} \in \mathbb{R}^8$ as the initialization. Let $x \in \mathbb{R}^2$ be the first two coordinates of v and $y \in \mathbb{R}^2$ be the third and fourth coordinate of v. Then, $x_0 = Z_{0,1:2} = x + \overline{x}$ and $y_0 = Z_{0,3:4} = y + \overline{y}$.

Because of the value of the initialization Z_0, we have for all $t \geqslant 0$, $Z_t = \Lambda^t Z_0 = \lambda^t v + \overline{\lambda}^t \overline{v}$, and thus $x_t = \lambda^t x + \overline{\lambda}^t \overline{x}$ and $y_t = \lambda^t y + \overline{\lambda}^t \overline{y}$.
This implies that the payoff at time t is (where $\langle \cdot, \cdot \rangle$ is the Hermitian inner product):

$$\begin{aligned} x_t^T A y_t &= \langle x_t, A y_t \rangle \\ &= \langle \lambda^t x + \overline{\lambda}^t \overline{x}, A(\lambda^t y + \overline{\lambda}^t \overline{y}) \rangle \\ &= \lambda^t \overline{\lambda}^t \langle x, Ay \rangle + \lambda^{2t} \langle x, A\overline{y} \rangle + \overline{\lambda}^{2t} \langle \overline{x}, Ay \rangle + \lambda^t \overline{\lambda}^t \langle \overline{x}, A\overline{y} \rangle \\ &= (\lambda\overline{\lambda})^t \langle x, Ay \rangle + (\lambda\overline{\lambda})^t \overline{\langle x, Ay \rangle} + \lambda^{2t} x^T A \overline{y} + \overline{\lambda^{2t} x^T A \overline{y}} \\ &= 2(\lambda\overline{\lambda})^t \Re(\langle x, Ay \rangle) + 2\Re(\lambda^{2t} \langle x, A\overline{y} \rangle) \ . \end{aligned}$$

Now, let us write the complex numbers λ, $x^T A y$ and $x^T A \overline{y}$ in there exponential form as $\lambda = \rho e^{i\alpha}$, $x^T A y = r e^{i\theta}$ and $x^T A \overline{y} = r' e^{i\theta'}$.
Then,

$$\begin{aligned} x_t^T A y_t &= 2(\rho e^{i\alpha} \rho e^{-i\alpha})^t 2\Re(r e^{i\theta}) + \Re(\rho^{2t} e^{2i\alpha t} r' e^{i\theta'}) \\ &= 2\rho^{2t} r \Re(e^{i\theta}) + 2\rho^{2t} r' \Re(e^{2i\alpha t + i\theta'}) \\ &= 2\rho^{2t} \left(r\cos(\theta) + r'\cos(2\alpha t + \theta') \right) \ . \end{aligned}$$

Assumption 2. Assume now that $r' > |r\cos(\theta)|$ and that α/π is not rational.

Let $\varepsilon > 0$ be such that $\varepsilon < |r'| - |r\cos(\theta)|$.
Due to the irrationality of α/π, $(\cos(\alpha t))_{t \in \mathbb{N}}$ is dense in [-1, 1].
Then, for all $T > 0$, there exist $t_1, t_2 \geqslant T$ such that $r'\cos(2\alpha t_1 + \theta') + r\cos(\theta) > \varepsilon$ and $r'\cos(2\alpha t_2 + \theta') + r\cos(\theta) < -\varepsilon$, for $\cos(2\alpha t_1 + \theta)$ close enough to 1 and $\cos(2\alpha t_2 + \theta)$ close enough to -1.
Thus, because $\rho > 1$, for all $M > 0$ and $T > 0$, there exist $t_1, t_2 \geqslant T$ such that $x_{t_1}^T A y_{t_1} > M$ and $x_{t_2}^T A y_{t_2} < -M$. This shows that the payoff is converging neither to a finite limit, nor to $+\infty$.

What still needs to be found for this proof to be true is matrices A and B such that Assumptions 1 and 2 are true.
Let $\nu \in \mathbb{C}$ be a square root of μ. We can see that for small values of η,

$$S_\eta^*(\mu) = \{1 + \eta\nu + o(\eta), 1 - \eta\nu + o(\eta), \eta\nu + o(\eta), -\eta\nu + o(\eta)\} \ .$$

Thus, for any ν with real value strictly positive, nonreal, and η small enough, Assumption 1 is true, and $\lambda = 1 + \eta\nu + o(\eta)$.
For example, let us choose $\mu = -3 + 4i$, and let $\nu = 1 + 2i$ be one of its square root.
We would like to find two matrices A and B in $\mathbb{R}^{2\times 2}$ such that $Sp(B^T A) = \{\mu, \overline{\mu}\}$. For this, we need that $\mathrm{Tr}(B^T A) = 2\Re(\mu)$ and $\det(B^T A) = |\mu|^2$.
Here, we could take $C = B^T A = \begin{pmatrix} -3 & -8 \\ 2 & -3 \end{pmatrix}$. We will fix the value of A and B later, we

just need for them to verify $B^TA = C$.
The eigenvalues of C are $\{\mu, \overline{\mu}\} = \{-3+4i, -3-4i\}$, and the spectrum of $\Lambda_{A,B}$ is $S^*_\eta(\mu) \cup S^*_\eta(\overline{\mu})$, with λ and $\overline{\lambda}$ the highest eigenvalues in absolute value of $\Lambda_{A,B}$, where $\lambda = \frac{1}{2}\left(1 + 2\eta\nu + \sqrt{1-4\eta^2\mu}\right)$.
We now assume that η is small enough such that λ and $\overline{\lambda}$ are the only eigenvalues ℓ of $\Lambda_{A,B}$ such that $|\ell| > 1$. Thus Assumption 1 is true.

The function $\eta \in \left[0, \frac{1}{2\sqrt{\mu_{\max}}}\right] \mapsto \frac{1}{2}\left(1 + 2\eta\nu + \sqrt{1-4\eta^2\mu}\right) \in \mathbb{C}$ is continuous, and takes nonreal values. Thus, there exist η small enough such that Assumption 1 is verified, and the argument α of λ is cannot be written as a rational factor times π.

We now would like to compare the values of x^TAy and $x^TA\overline{y}$ to check if the Assumption 2 is also verified.
For $\mu = -3+4i$, we have that y' is an eigenvector of B^TA with eigenvalue μ is and only if there exist a scalar $c \in \mathbb{C}$ such that $y' = c\begin{pmatrix} 2i \\ 1 \end{pmatrix}$. Let us choose y' such that $y = \lambda y' = \begin{pmatrix} 2i \\ 1 \end{pmatrix}$.
We have:

$$r\cos(\theta) = \Re\left(\langle x, Ay\rangle\right) = \Re\left(\frac{1}{\nu}\langle Ay, Ay\rangle\right) = \Re\left(\frac{1}{\nu}\|Ay\|^2\right),$$

and

$$r' = |\langle x, A\overline{y}\rangle| = \left|\frac{1}{\nu}\langle Ay, A\overline{y}\rangle\right| = \frac{|\langle Ay, A\overline{y}\rangle|}{|\nu|}.$$

What we need to do is to find a matrix A invertible such that $r' > r\cos(\theta)$, and setting $B = (CA^{-1})^T$.
For $A = \begin{pmatrix} 3 & 8 \\ -2 & 3 \end{pmatrix}$, we get $B = -I_d$. Then

$$r\cos(\theta) = \Re(25-50i) = 25 \text{ and } r' = |33+6i| > 33 > r\cos(\theta).$$

Thus, Assumption 2 is also verified.
This proves the existence of some general-sum game (A, B), and of some $\eta < \frac{1}{2\sqrt{\mu_{\max}}}$ such that for some initialization vector, we are not converging to a Nash equilibrium, nor to infinite payoff for both players.

□

Chapter 4

Certified Multi-Fidelity Zeroth-order Optimization

We consider the problem of multi-fidelity zeroth-order optimization, where one can evaluate a function f at various approximation levels (of varying costs), and the goal is to optimize f with the cheapest evaluations possible. In this paper, we study *certified* algorithms, which are additionally required to output a data-driven upper bound on the optimization error. We first formalize the problem in terms of a min-max game between an algorithm and an evaluation environment. We then propose a certified variant of the MFDOO algorithm and derive a bound on its cost complexity for any Lipschitz function f. We also prove an f-dependent lower bound showing that this algorithm has a near-optimal cost complexity. We close the paper by addressing the special case of noisy (stochastic) evaluations as a direct example.

This chapter is based on a joint work with Sébastien Gerchinovitz.

Contents

4.1 Introduction 90
4.1.1 Setting 90
4.1.2 Main contributions and outline of the chapter 92
4.1.3 Notation 92
4.2 The certified MFDOO algorithm, and its cost complexity . . . 93
4.3 Lower bound 99
4.4 Special case: noisy evaluations of f (a.k.a. stochastic setting) . 103
4.A Missing proofs in Section 4.3 107
4.B Useful lemmas 109
4.C The special case of constant costs 110
4.D On the characterization of $S_\beta(f,\varepsilon)$ 111

4.1 Introduction

We consider the problem of multi-fidelity zeroth-order optimization, which unfolds roughly as follows (details are given in Section 4.1.1). Let $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ be a function. Assume that at any $x \in \mathcal{X}$, we can query the value $f(x)$ with any desired accuracy $\alpha > 0$, at a cost of $c(\alpha)$. Accurate evaluations (small α) come at a high cost. The goal of multi-fidelity optimization is to maximize f with the cheapest evaluations possible.

A typical example is the optimization of a function f computed with finite element modeling. A case with two fidelity functions (two values of α) appears in Sun et al. (2011), for sheet-metal forming design with the goal of having no defects in the products (automobile inner panel in that paper). Given three variables x_1, x_2, x_3 modeling strong restraining forces on the metal, the goal is to set this forces to a good value to avoid both rupture and wrinkling. Two different finite element solvers were used to approximate f at any point x: incremental finite element solvers, or a one-step finite element model, which is computationally cheap but provides worse estimates than the former model. Finding a good design of the forces at a reasonable computational cost is an example of multi (two) fidelity optimization problem. Many other examples can be found, e.g., in thermodynamics Dewettinck et al. (1999); Le Gratiet (2013), design of new aircraft Geiselhart et al. (2011), or nuclear criticality safety Picheny et al. (2010).

Certified optimization In practice, algorithms that achieve small optimization errors with small evaluation costs are desirable but may not inform the user when a small optimization error has been obtained. In the example above, an engineer might require to *certify* the output of the algorithm, that is, to get a guaranteed optimization error bound that they can compute by only using the observed data and some (light) prior knowledge on f, as is done, e.g., in Hansen et al. (1992a); Bachoc et al. (2021) in the single-fidelity setting. Such requirement can be important in industrial fields involving safety-critical systems (e.g., cars, aircraft, health, nuclear engineering).

In this paper, we study the problem of finding a *certifiably* approximate-maximizer of a Lipschitz function f in the multi-fidelity setting. We quantify the smallest evaluation cost to reach this goal, by deriving nearly-matching upper and lower bounds for any such f.

4.1.1 Setting

We now formally define the setting. Let $\mathcal{X} \subset \mathbb{R}^n$ be a non-empty[1] compact set endowed with a norm $\|\cdot\|$, and $f : \mathcal{X} \to \mathbb{R}$ be an L-Lipschitz function, with a maximizer $x^\star \in \mathcal{X}$.[2] Let also $c : (0, +\infty) \to [0, +\infty)$ be a non-increasing cost function.

The problem, which we describe in the online protocol below, can be seen as an interaction between two players: the algorithm A whose goal is to maximize f, and an environment E which returns perturbed values of f. They interact together in the following way: at every round $t \geqslant 1$, A picks a query point $x_t \in \mathcal{X}$ and an evaluation accuracy $\alpha_t > 0$; it then observes $r_t = E_t(x_t, \alpha_t) \in [f(x_t) - \alpha_t, f(x_t) + \alpha_t]$ at a cost of $c(\alpha_t)$ ($E_t(\cdot, \alpha)$

[1] All throughout the paper, $\mathcal{X}$ is implictly assumed to be non empty.

[2] In fact, all results of Section 4.2 still hold if f is simply L-Lipschitz around $x^\star$, that is, if $f(x) \geqslant f(x^\star) - L\|x - x^\star\|$ for all $x \in \mathcal{X}$.

is sometimes called the α-fidelity function); finally A recommends some candidate $x_t^* \in \mathcal{X}$ for a maximizer of f, and outputs an error certificate $\xi_t \geqslant 0$ with the constraint that $\xi_t \geqslant \max_{x\in\mathcal{X}} f(x) - f(x_t^*)$ (see a more formal definition below). This way, when using the algorithm, we can not only find an ε-maximizer x_t^*, but we can *know* when it is ε-optimal by looking at ξ_t, and thus confidently stop searching.

Note that the case of the constant cost $c(\alpha) = 1$ for all $\alpha > 0$ can be reduced to the single-fidelity setting, where f is observed perfectly (see Appendix 4.C for details).

Online Protocol: Certified multi-fidelity zeroth-order optimization

---

Init: The environment secretly observes f and picks $E = (E_t)_{t\geqslant 1}$ with $E_t(x,\alpha) \in \big[f(x)-\alpha, f(x)+\alpha\big]$ for all $t \geqslant 1$, $x \in \mathcal{X}$, and $\alpha > 0$ (we also call E *the environment*)

1: **for** $t = 1, 2, \ldots,$ **do**
2: A chooses a query point $x_t \in \mathcal{X}$ and an evaluation accuracy $\alpha_t > 0$
3: A incurs a cost $c(\alpha_t)$
4: E returns $r_t = E_t(x_t, \alpha_t) \in [f(x_t) - \alpha_t, f(x_t) + \alpha_t]$ (inaccurate evaluation of $f(x_t)$)
5: A returns a recommendation x_t^* for the maximum of f, with an error certificate $\xi_t \geqslant 0$
6: **end for**

Next we introduce key definitions before describing the optimization goal, our contributions, related works, and useful notation.

Definitions: environments and certified algorithms For any L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$, we define the set $\mathcal{E}(f)$ of all environments associated with f, which are sequences of functions $E = (E_t)_{t\geqslant 1}$ with $E_t(x,\alpha) \in \big[f(x)-\alpha, f(x)+\alpha\big]$ for all $t \geqslant 1$, $x \in \mathcal{X}$ and $\alpha > 0$. We assume for simplicity that the sequence $E \in \mathcal{E}(f)$ is fixed from the beginning of the online protocol.[3]

We can now formally define *certified algorithms*. As can be seen from the online protocol above, x_t and α_t are deterministic functions of the past observations $r_1, \ldots, r_{t-1}$, while x_t^* and ξ_t are deterministic functions of $r_1, \ldots, r_t$. We additionally require that the certificates ξ_t satisfy $\xi_t \geqslant \max_{x\in\mathcal{X}} f(x) - f(x_t^*)$ for all rounds $t \in \mathbb{N}^*$, all L-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$ and all environments $E \in \mathcal{E}(f)$. We call *certified algorithm* any such sequence of functions $A = \big(x_t(\cdot), \alpha_t(\cdot), x_t^*(\cdot), \xi_t(\cdot)\big)_{t\geqslant 1}$, and let $\mathcal{A}$ denote the set of all certified algorithms. With a slight abuse of notation, we also sometimes write $x_t(E)$, $\alpha_t(E)$, $x_t^*(E)$ and $\xi_t(E)$ to make the dependency on E more explicit.

Optimization goal Recall that $x^\star \in \mathcal{X}$ denotes a maximizer of f. A classic goal in multi-fidelity optimization is to reach a small optimization error $f(x^\star) - f(x_t^*)$ while minimizing the total cost $\sum_{s=1}^t c(\alpha_s)$ (see, e.g., Sen et al. (2018, 2019); Fiegel et al. (2020)). In this paper, we address the stronger goal of finding a recommendation x_t^* with an error certificate ξ_t below ε (and thus an optimization error *known* to be bounded by ε), with the smallest cumulative cost $\sum_{s=1}^t c(\alpha_s)$ possible.

[3] Since we only consider deterministic algorithms A and work towards guarantees that hold uniformly over all environments, this is in fact equivalent to playing against adversarial environments.

More formally, for any environment $E \in \mathcal{E}(f)$, we define the *cost complexity* $\sigma(A, E, \varepsilon)$ as the smallest total cost for which A can output a certificate below ε (when run against E). It can be expressed as follows (by convention, $\sigma(A, E, \varepsilon) = +\infty$ if no such C exists):

$$\sigma(A, E, \varepsilon) = \inf \left\{ C \in \mathbb{R} : \exists \tau \in \mathbb{N}^*, \sum_{t=1}^{\tau} c(\alpha_t(E)) \leqslant C \text{ and } \xi_\tau(E) \leqslant \varepsilon \right\}. \tag{4.1}$$

Equivalently, since costs are nonnegative, $\sigma(A, E, \varepsilon)$ is equal to the total cost incurred by A (when run against E) until its certificate ξ_t falls below ε for the first time.

In this paper, we are interested in certified algorithms with small cost complexity against any environment, that is, in algorithms $A \in \mathcal{A}$ that approximately reach the infimum

$$\inf_{A \in \mathcal{A}} \sup_{E \in \mathcal{E}(f)} \sigma(A, E, \varepsilon)$$

for any unknown L-Lipschitz function f. Importantly, the above min-max quantity depends on f (through the set $\mathcal{E}(f)$), since some functions are easier than others to maximize with certified algorithms.

4.1.2 Main contributions and outline of the chapter

In this paper, we study the cost complexity of certified algorithms to maximize L-Lipschitz functions in the multi-fidelity setting. We prove nearly-matching f-dependent upper and lower bounds, which extend the single-fidelity results of Bachoc et al. (2021). More precisely, we make the following set of contributions.

- On the modeling side, we formalize the problem of *certified* multi-fidelity zeroth-order optimization (see Section 4.1.1 above).
- In Section 4.2 we define a certified variant of the MFDOO algorithm Sen et al. (2018). We bound its cost complexity in terms of the key quantity $S_\beta(f, \varepsilon)$ defined in (4.2) below.
- In Section 4.3 we derive a nearly-matching f-dependent lower bound that holds for all certified algorithms (i.e., we derive a lower bound on the min-max quantity above).
- Finally, in Section 4.4 we address the special case of noisy evaluations of f. By a simple reduction to the (deterministic) multi-fidelity setting above, we derive a high-probability sample complexity bound which solves a conjecture by Bachoc et al. (2021).

In the appendix we collect some technical proofs, useful simple geometric lemmas, together with a formal intuitive reduction from the single- to the multi-fidelity setting (Appendix 4.C).

4.1.3 Notation

We collect below some notation that is used all throughout the paper, including the key quantity $S_\beta(f, \varepsilon)$ defined in (4.2) below.

Standard notation $\mathbb{N} = \{0,1,2,\dots\}$ denotes the set of natural numbers, and $\mathbb{N}^* = \mathbb{N}\backslash\{0\}$ denotes the set of positive natural numbers. For any $x \in \mathbb{R}^n$ and $\rho > 0$, we write $B(x,\rho) = \{u \in \mathbb{R}^n : \|u - x\| \leqslant \rho\}$ for the closed ball centered at x with radius ρ.

Lipschitz functions, ε-optimal points, layers Let $\mathcal{F}_L$ denote the set of L-Lipschitz functions from $\mathcal{X}$ to $\mathbb{R}$. Let also $\mathrm{diam}(\mathcal{X}) = \sup_{x,y\in\mathcal{X}} \|x - y\|$ denote the diameter of $\mathcal{X}$. Since f is L-Lipschitz, the largest possible optimization error $f(x^\star) - f(x_t^*)$ is bounded by $\varepsilon_0 := L \cdot \mathrm{diam}(\mathcal{X})$. In the sequel, we will thus only consider values $\varepsilon \in (0, \varepsilon_0)$, and set $m_\varepsilon := \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$. For any $1 \leqslant k \leqslant m_\varepsilon - 1$ we define the intermediate target error $\varepsilon_k := \varepsilon_0 2^{-k}$; we also set $\varepsilon_{m_\varepsilon} := \varepsilon$.

For any $0 \leqslant a < b$, we denote the set of a-optimal points by $\mathcal{X}_a := \{x \in \mathcal{X} : f(x^\star) - f(x) \leqslant a\}$, and we define the *layer* $\mathcal{X}_{(a,b]} := \{x \in \mathcal{X} : a < f(x^\star) - f(x) \leqslant b\}$, which is the set of b-optimal points that are not a-optimal.

Packing number For any $r > 0$, the r-packing number $\mathcal{N}(\mathcal{X}', r)$ of a subset $\mathcal{X}' \subset \mathcal{X}$ is the largest number k of r-separated points $x'_1, \dots, x'_k \in \mathcal{X}'$, that is, such that $\left\|x'_i - x'_j\right\| > r$ for all $i \neq j \leqslant k$. (By convention, $\mathcal{N}(\mathcal{X}', r) = 0$ if $\mathcal{X}'$ is empty. Note also that $\mathcal{N}(\mathcal{X}', r) < +\infty$ since $\mathcal{X}$ is compact.)

The complexity quantity $S_\beta(f,\varepsilon)$ For any $\beta > 0$, we set

$$S_\beta(f,\varepsilon) := \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) c(\beta\varepsilon) + \sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) c(\beta\varepsilon_k) \ . \tag{4.2}$$

As we will see later, $S_\beta(f,\varepsilon)$ plays a key role in the optimal cost complexity of certified algorithms. We briefly explain why. Let $x \in \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1})}$. To realize that x belongs to the layer $\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1})}$, a single call to f at x with (prophetic) evaluation accuracy $\alpha \approx \varepsilon_k$ would be enough, yielding a cost roughly of $c(\varepsilon_k)$. By L-Lipschitz continuity of f, this single evaluation also helps classify nearby points (at distance roughly ε_k/L) within the same layer. Therefore, the whole layer could (hopefully) be identified with roughly $\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right)$ evaluations of f at accuracy $\alpha \approx \varepsilon_k$. Repeating these arguments over all layers and using another (similar) argument over $\mathcal{X}_\varepsilon$ suggests that the sum $S_\beta(f,\varepsilon)$ above characterizes the optimal cost complexity of certified algorithms. Next we prove upper and lower bounds supporting this intuition.

4.2 The certified MFDOO algorithm, and its cost complexity

In this section we define a certified version of the MFDOO algorithm Sen et al. (2018), and then study its cost complexity (Theorem 4.2 below), which we will prove to be nearly optimal in Section 4.3.

Similarly to MFDOO Sen et al. (2018) and its ancestors (e.g., the branch-and-bound algorithm of Perevozchikov (1990), HOO Bubeck et al. (2011a), DOO Munos (2011), POO Grill et al. (2015) c.DOO Bachoc et al. (2021), etc), our algorithm takes as input

a hierarchical partitioning of $\mathcal{X}$, that is, a tree-based structure X in which each node represents a region of $\mathcal{X}$ and has K children, which correspond to a K-partition of the parent region. More precisely, X is an infinite sequence of subsets $(X_{h,i})_{h\in\mathbb{N},i\in\{0,\dots,K^h-1\}}$ of $\mathcal{X}$ called *cells* such that $\mathcal{X}\subset X_{0,0}$, and for any depth $h\in\mathbb{N}$ and location $0\leqslant i\leqslant K^h-1$, the cells $X_{h+1,Ki},\dots,X_{h+1,K(i+1)-1}$ form a partition of $X_{h,i}$ (the nodes $(h+1,Ki),\dots,(h+1,K(i+1)-1)$ are the *children* of node (h,i)). Each cell has a *representative* $x_{h,i}\in X_{h,i}$. We assume that $x_{h,i}\in\mathcal{X}$ whenever $X_{h,i}\cap\mathcal{X}\neq\varnothing$. (A typical example is the barycenter of the cell, if inside $\mathcal{X}$.)

In all the chapter, we make the following two assumptions. The first one is classical (e.g., Bubeck et al. (2011a); Munos (2011)). The second one appeared in Bachoc et al. (2021) and was useful to derive bounds on DOO or c.DOO in terms of packing numbers of layers $\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}$. Both assumptions can always be satisfied when $\mathcal{X}$ is compact. For example, if $\mathcal{X}=[0,1]^n$ and $\|\cdot\|$ is the sup norm, we can take the regular dyadic partitioning $(X_{h,i})_{h\in\mathbb{N},i\in\{0,\dots,2^{nh}-1\}}$ consisting of 2^{nh} cubes of size 2^{-h} at depth $h\geqslant 0$, with centers $x_{h,i}$. Then, Assumptions 3 and 4 hold true with $R=1$ and $\delta=\nu=1/2$.

Assumption 3. There exist two positive constants $\delta\in(0,1)$ and $R>0$ such that, for all $h\in\mathbb{N}$, $i\in\{0,\dots,K^h-1\}$, and all $u,v\in X_{h,i}$, we have $\|u-v\|\leqslant R\delta^h$.

Assumption 4. There exists $\nu>0$ such that, with δ as in Assumption 3, for any $h,h'\in\mathbb{N}$, $i\in\{0,\dots,K^h-1\}$ and $i'\in\{0,\dots,K^{h'}-1\}$ with $(h,i)\neq(h',i')$, we have $\|x_{h,i}-x_{h',i'}\|\geqslant\nu\delta^{\max\{h,h'\}}$.

We now define c.MF-DOO (Certified Multi-Fidelity Deterministic Optimistic Optimization). The pseudo-code is given in Algorithm 11 below. The algorithm maintains a set $\mathcal{L}_t$ of active nodes (or *leaves*) whose associated cells cover $\mathcal{X}$. At the end of each iteration (Line 16), c.MF-DOO picks the most promising leaf (h^*,i^*) by maximizing the surrogate $r_{h,i}+LR\delta^h+\alpha_{h,i}$, which is an upper bound on $f(x)$ for any $x\in X_{h,i}$.[4] Then, during the next iteration, c.MF-DOO develops the tree by querying one after the other all the (feasible) children of (h^*,i^*). At Line 9, it picks $x_t=x_{h^*+1,j}$ as the next query point and $\alpha_t=LR\delta^{h^*+1}$ for the accuracy. (A much smaller value of α_t could be counterproductive: it could come at a much higher cost, while not improving the optimization process by much, since the surrogate is an over-approximation of f with a mistake possibly of the order of $LR\delta^{h^*+1}$ on the cell $X_{h^*+1,j}$ even if f were observed exactly.) After receiving the approximate evaluation y_t of $f(x_t)$, c.MF-DOO returns the recommendation $x_t^*=x_{\tilde{t}}$ (see Line 11), which is the point with the currently best guaranteed value of f. Finally, the certificate ξ_t at Lines 12 or 17 is the difference between a guaranteed upper bound $r_{h^*,i^*}+LR\delta^{h^*}+\alpha_{h^*,i^*}$ on $\max(f)$ and a guaranteed lower bound $r_{\tilde{t}}-\alpha_{\tilde{t}}$ on $f(x_t^*)$.

Note that in Algorithm 11 and in the rest of the paper, we identify any round $t\geqslant 1$ with the node (h,i) that is queried at time t.[5] Depending on our needs, we index quantities either by rounds or nodes (writing, e.g., x_t or $x_{h,i}$).

Before bounding the cost complexity of c.MF-DOO, we start by proving that the ξ_t's defined at Lines 3, 12 and 17 are valid certificates.

[4] Indeed, $r_{h,i}$ is an $\alpha_{h,i}$-approximation of $f(x_{h,i})$, f is L-Lipschitz, and the maximum distance from a point in $X_{h,i}$ to $x_{h,i}$ is at most $R\delta^h$, so that $\max_{x\in X_{h,i}}f(x)\leqslant f(x_{h,i})+LR\delta^h\leqslant r_{h,i}+LR\delta^h+\alpha_{h,i}$.

[5] By definition of Algorithm 11, there is indeed an injection $t\in\mathbb{N}^*\mapsto(h,i)$ with $h\in\mathbb{N},i\in\{0,\dots,K^h-1\}$.

Algorithm 11 c.MF-DOO (Certified Multi-Fidelity Deterministic Optimistic Optimization)

Inputs: $\mathcal{X}$, K, $(X_{h,i})_{h\in\mathbb{N},i\in\{0,\dots,K^h-1\}}$, $(x_{h,i})_{h\in\mathbb{N},i\in\{1,\dots,K^h-1\}}$, δ, R, L
Initialization Let $t \leftarrow 1$ and $\mathcal{L}_1 \leftarrow \{(0,0)\}$
1: Pick the first query point $x_1 \leftarrow x_{0,0}$, and the first accuracy $\alpha_1 \leftarrow LR$
2: Observe the value $r_1 = E_1(x_1,\alpha_1) \in [f(x_1)-\alpha_1, f(x_1)+\alpha_1]$
3: Output the recommendation $x_1^* \leftarrow x_1$ and certificate $\xi_1 \leftarrow LR$
4: Pick the first node $(h^*,i^*) \leftarrow (0,0)$
5: **for** *iteration* $= 1,2,\dots$ **do**
6: **for all** child (h^*+1,j) of (h^*,i^*) **do**
7: **if** $X_{h^*+1,j} \cap \mathcal{X} \neq \varnothing$ **then**
8: Let $t \leftarrow t+1$ and $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{(h^*+1,j)\}$
9: Pick the query point $x_t \leftarrow x_{h^*+1,j}$ and accuracy $\alpha_t \leftarrow LR\delta^{h^*+1}$
10: Observe the value $r_t = E_t(x_t,\alpha_t) \in [f(x_t)-\alpha_t, f(x_t)+\alpha_t]$ given by E
11: Output the recommendation $x_t^* = x_{\widetilde{t}}$, with $\widetilde{t} \in \arg\max_{1\leqslant s\leqslant t}\{r_s-\alpha_s\}$
12: Output the certificate $\xi_t = r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} - (r_{\widetilde{t}}-\alpha_{\widetilde{t}})$
13: **end if**
14: **end for**
15: Remove (h^*,i^*) from $\mathcal{L}_t$
16: Let $(h^*,i^*) \in \arg\max_{(h,i)\in\mathcal{L}_t}\{r_{h,i}+LR\delta^h+\alpha_{h,i}\}$
17: Update the last certificate $\xi_t = r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} - (r_{\widetilde{t}}-\alpha_{\widetilde{t}})$
18: **end for**

Lemma 4.1. *Suppose Assumption 3 holds, and that $f:\mathcal{X}\to\mathbb{R}$ is an L-Lipschitz function, with a maximizer $x^\star \in \mathcal{X}$. Then, for any environment $E \in \mathcal{E}(f)$ and any $t \in \mathbb{N}^*$, the quantity ξ_t defined at Lines 3, 12 and 17 of Algorithm 11 is a valid certificate, that is: $f(x^\star) - f(x_t^*) \leqslant \xi_t$.*

Proof. Since f is L-Lipschitz and $R \geqslant \operatorname{diam}(\mathcal{X})$, note that $\xi_1 = LR \geqslant f(x^\star) - f(x_1^*)$. We now prove the lemma for any subsequent round. For any $t' \geqslant 2$, consider the moment when the algorithm reaches Line 12 with $t = t'$. Next we show that the certificate $\xi_t = r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} - \max_{s\leqslant t}(r_s-\alpha_s)$ defined at that time satisfies $\xi_t \geqslant f(x^\star) - f(x_t^*)$ (the potential update at Line 17 will be addressed at the end of the proof). The associated node (h^*,i^*) was either defined at Line 4 if the outer `for` loop is still at iteration 1, or at Line 16 otherwise. In both cases, we have $(h^*,i^*) \in \arg\max_{(h,i)\in\mathcal{L}_m}\{r_{h,i}+LR\delta^h+\alpha_{h,i}\}$ for a number $m \leqslant t-1$ of evaluations of f.

Note that, by induction on the `iteration` variable, the cells $X_{h,i}$ associated with the leaves $(h,i) \in \mathcal{L}_m$ form a partition of a superset of $\mathcal{X}$. Let $(\bar{h},\bar{i}) \in \mathcal{L}_m$ be the node of the cell $X_{\bar{h},\bar{i}}$ containing $x^\star$. By the maximizing property of (h^*,i^*), we have

$$\begin{aligned} r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} &\geqslant r_{\bar{h},\bar{i}} + LR\delta^{\bar{h}} + \alpha_{\bar{h},\bar{i}} \\ &\geqslant f(x_{\bar{h},\bar{i}}) + LR\delta^{\bar{h}} \geqslant f(x^\star)\,, \end{aligned} \tag{4.3}$$

where the last line follows from $|r_{\bar{h},\bar{i}} - f(x_{\bar{h},\bar{i}})| \leqslant \alpha_{\bar{h},\bar{i}}$, Assumption 3, and the fact that f is L-Lipschitz. To conclude, note that $r_s - \alpha_s \leqslant f(x_s)$ for all $s \leqslant t$, and therefore

$r_{\tilde{t}} - \alpha_{\tilde{t}} \leqslant f(x_{\tilde{t}}) = f(x_t^*)$. Combining this with (4.3) entails that $\xi_t \geqslant f(x^\star) - f(x_t^*)$. Noting that the same arguments apply (with $m = t$) if ξ_t is redefined at Line 17 concludes the proof. □

We just proved that c.MF-DOO is a certified algorithm. We now show that its cost complexity can be controlled in terms of the quantity $S_\beta(f,\varepsilon)$ defined in (4.2). We recall that $\varepsilon_0 := L\,\mathrm{diam}(\mathcal{X})$, $m_\varepsilon := \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$, $\varepsilon_{m_\varepsilon} := \varepsilon$, and $\varepsilon_k := \varepsilon_0 2^{-k}$ for all $1 \leqslant k \leqslant m_\varepsilon - 1$.

Theorem 4.2. *Assume that $c : \mathbb{R}^+ \to \mathbb{R}$ is non-increasing and that $\mathcal{X} \subset \mathbb{R}^d$ is compact. Suppose that Assumptions 3 and 4 hold, and that for some known $L > 0$, c.MF-DOO is run with evaluation accuracies $\alpha_{h,i} = LR\delta^h$ (cf. Algorithm 11). Then there exists a constant $a > 0$ (e.g., $a = K$ if $\nu \geqslant 3R$ or $a = K(1+6R/\nu)^n$ otherwise) such that, for any L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$, any environment $E \in \mathcal{E}(f)$, and any $\varepsilon \in (0, \varepsilon_0)$,*

$$\sigma(c.MF\text{-}DOO, E, \varepsilon) \leqslant a S_{\frac{\delta}{3}}(f,\varepsilon) + c(LR)\,.$$

We make several comments before proving the theorem.

Related upper bounds A similar upper bound was proved in (Bachoc et al., 2021, Theorem 1) when f is evaluated perfectly, which corresponds to the special case $c(\alpha) = 1$ for all α. Theorem 4.2 above generalizes this result (up to constants) to the multi-fidelity setting.

Other related results are bounds for MFDOO Sen et al. (2018) and Kometo Fiegel et al. (2020), which are multi-fidelity algorithms *without certificates*. In that setting, the performances are measured differently. The cost complexity can be defined as the total cost incurred by the algorithm before outputting an ε-optimal recommendation (the difference with the certified setting is that the learner has no *observable* proof that an ε-maximizer has been found.) With such performance measure, MFDOO satisfies a complexity bound similar to $S_\beta(f,\varepsilon)$ but *without the first term* $\mathcal{N}(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L})\, c(\beta\varepsilon)$ in (4.2). [6] This difference can be negligible for some functions (e.g., if $c_1 \|x - x^\star\|^\nu \leqslant f(x^\star) - f(x) \leqslant c_2 \|x - x^\star\|^\nu$ for all $x \in \mathcal{X}$ and some $c_1, c_2 > 0$ and $0 < \nu \leqslant 1$, where $x^\star$ is a maximizer of f), but it can be dramatic for other functions. For instance, for constant functions, the term $\mathcal{N}(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L})\, c(\beta\varepsilon)$ is of the order of $(L/\varepsilon)^n c(\beta\varepsilon)$. The reason behind this large additional term in the certified setting is intuitive: a constant function f is perfectly optimized after one evaluation only, but *certifying* the result at accuracy ε somehow requires to evaluate the function on a ε/L-cover of $\mathcal{X}$ with accuracies $\alpha_t \approx \varepsilon$, so as to make sure no bumps of size ε were forgotten.

Note also that, contrary to Fiegel et al. (2020), we work with a *known* bias function but an *unknown* cost function. This is because we aim at *certifying* an ε-maximizer of f, rather than optimally allocating a total evaluation budget Λ.

[6] This bound can be proved along the same lines as those of Theorem 4.2. See also (Sen et al., 2018, Theorem 1) and (Fiegel et al., 2020, Theorem 3) for similar bounds under slightly weaker assumptions (relating f directly to the hierarchical partitioning) but that are expressed in terms of a near-optimality dimension of f, and thus do not reflect the fact that constant functions are easy to optimize. Roughly speaking, the bound of (Sen et al., 2018, Theorem 1) for MFDOO is in spirit close to $\sum_{k=1}^{m_\varepsilon} \mathcal{N}\big(\mathcal{X}_{\varepsilon_{k-1}}, \varepsilon_k/L\big) c(\beta\varepsilon_k)$, instead of the tighter bound $\sum_{k=1}^{m_\varepsilon} \mathcal{N}\big(\mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]}, \varepsilon_k/L\big) c(\beta\varepsilon_k)$.

On the choice of ε_k As can be seen from the proof below, the upper bound is actually true for any decreasing sequence $\varepsilon_0 = L\operatorname{diam}(\mathcal{X}) > \varepsilon_1 > \ldots > \varepsilon_{m-1} > \varepsilon_m = \varepsilon$ and any $m \geqslant 1$. The specific sequence $\varepsilon_k := \varepsilon_0 2^{-k}$ however realizes a good trade-off between small ratios $\varepsilon_{k-1}/\varepsilon_k \leqslant 2$ and a small number of terms $m_\varepsilon = \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$. The nearly-matching lower bound of Section 4.3 will indeed imply that this sequence is nearly optimal.

Possible improvements or consequences Note that the constant a was not optimized and could likely be improved. Furthermore, similarly to Bachoc et al. (2021), under a mild geometric condition on $\mathcal{X}$ recalled in Section 4.4, the sum $S_{\delta/3}(f,\varepsilon)$ can be bounded (up to multiplicative constants) in between two integrals of the form $\int_{\mathcal{X}} c(b\cdot(\Delta_x+\varepsilon))/(\Delta_x+\varepsilon)^n \,\mathrm{d}x$, where $\Delta_x = f(x^\star) - f(x)$ and $b \in \{\delta/3, \delta/12\}$ (provided $\varepsilon < \varepsilon_0/2$). This integral is proven in Lemma 4.13 in the appendix.

Proof of Theorem 4.2. The proof generalizes that of (Bachoc et al., 2021, Theorem 1) to the multi-fidelity setting, with similar arguments yet a few technical subtleties. In order to bound the total cost incurred by c.MF-DOO against environment E, we control the index $I_\varepsilon \geqslant 1$ of the first iteration (cf Line 5) at the end of which the certificate falls below ε. More precisely, let (h^*_ℓ, i^*_ℓ) be the node chosen at the end of each iteration $\ell \geqslant 1$ (Line 16). Then, we define I_ε by[7]

$$I_\varepsilon = \inf\Big\{ \ell \in \mathbb{N}^* : r_{h^*_\ell, i^*_\ell} + LR\delta^{h^*_\ell} + \alpha_{h^*_\ell, i^*_\ell} \leqslant \max_{s \leqslant T_\ell} \{r_s - \alpha_s\} + \varepsilon \Big\} \,,$$

where for any $\ell \geqslant 1$, the quantity T_ℓ denotes the total number of evaluations of f until the leaf (h^*_ℓ, i^*_ℓ) is selected at Line 16. Next we focus on $\tau := T_{I_\varepsilon}$. Note that $\xi_\tau \leqslant \varepsilon$ by definition of I_ε and ξ_τ (in Line 17). Recalling that $\sigma(\text{c.MF-DOO}, E, \varepsilon)$ is the total cost that c.MF-DOO incurs until outputting a certificate below ε for the first time, this entails

$$\sigma(\text{c.MF-DOO}, E, \varepsilon) \leqslant \sum_{t=1}^{\tau} c(\alpha_t) \,. \tag{4.4}$$

We now split the right-hand side into several terms involving the layers $\mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]}$. We set $(h^*_0, i^*_0) = (0,0)$. Note that the points x_t queried at times $t \in \{2, \ldots, \tau\}$ are all associated with nodes (h,i) that are children of some (h^*_ℓ, i^*_ℓ), $\ell = 0, \ldots, I_\varepsilon - 1$, and that these (h,i) are queried only once. Therefore,

$$\sum_{t=1}^{\tau} c(\alpha_t) \leqslant c(\alpha_1) + \sum_{\ell=0}^{I_\varepsilon - 1} \sum_{j=Ki^*_\ell}^{K(i^*_\ell+1)-1} c(\alpha_{h^*_\ell+1, j}) = c(LR) + K \sum_{x_{h^*, i^*} \in \mathcal{E}_\varepsilon} c(LR\delta^{h^*+1}) \,, \tag{4.5}$$

where we set $\mathcal{E}_\varepsilon := \{x_{h^*_0, i^*_0}, \ldots, x_{h^*_{I_\varepsilon - 1}, i^*_{I_\varepsilon - 1}}\}$ (the $x_{h,i}$ are pairwise-distinct by Assumption 4).

We now split the sum over $\mathcal{E}_\varepsilon$ above into $m_\varepsilon + 1 = \lceil \log_2(\varepsilon_0/\varepsilon) \rceil + 1$ terms. Recall from Section 4.1.3 that $\varepsilon_0 = L\operatorname{diam}(\mathcal{X})$, $\varepsilon_k = \varepsilon_0 2^{-k}$ for $1 \leqslant k \leqslant m_\varepsilon - 1$, and $\varepsilon_{m_\varepsilon} = \varepsilon$. Since the sets $\mathcal{X}_\varepsilon$ (all ε-optimal points) and $\mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]}$, $k = 1, \ldots, m_\varepsilon$ (all points in between ε_k and ε_{k-1} optimal) form a partition of $\mathcal{X}$,

$$\mathcal{E}_\varepsilon = (\mathcal{E}_\varepsilon \cap \mathcal{X}_\varepsilon) \cup \bigcup_{k=1}^{m_\varepsilon} \Big(\mathcal{E}_\varepsilon \cap \mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]} \Big) \,. \tag{4.6}$$

[7] The rest of the proof implies that the set is never empty, so that $I_\varepsilon < +\infty$.

Let $N_{\varepsilon,k}$ be the cardinality of $\mathcal{E}_\varepsilon \cap \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}$ for all $1 \leqslant k \leqslant m_\varepsilon$ and $N_{\varepsilon,m_\varepsilon+1}$ be the cardinality of $\mathcal{E}_\varepsilon \cap \mathcal{X}_\varepsilon$. Moreover, let $h_{\varepsilon,k}$ be the maximum depth h^* reached by points x_{h^*,i^*} in $\mathcal{E}_\varepsilon \cap \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}$ for $1 \leqslant k \leqslant m_\varepsilon$, and $h_{\varepsilon,m_\varepsilon+1}$ be the maximum depth reached by points in $\mathcal{E}_\varepsilon \cap \mathcal{X}_\varepsilon$. By (4.5), (4.6), and the fact that $\alpha \mapsto c(\alpha)$ is non-increasing, we have:

$$\sum_{t=1}^{\tau} c(\alpha_t) \leqslant c(LR) + K \sum_{k=1}^{m_\varepsilon+1} N_{\varepsilon,k}\, c(LR\delta^{h_{\varepsilon,k}+1}) \,. \tag{4.7}$$

We now bound $N_{\varepsilon,k}$ from above and $LR\delta^{h_{\varepsilon,k}+1}$ from below (see (4.9), (4.10), (4.11), and (4.12)).

We start by proving (4.8) below. Let $x^\star \in \mathcal{X}$ be a maximizer of f. Following the same arguments as before (4.3) (using f being L-Lipschitz and $E \in \mathcal{E}(f)$), we can see that, for any node (h^*, i^*) selected at Line 16,

$$r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} \geqslant f(x^\star) \,.$$

This implies that $f(x_{h^*,i^*}) + LR\delta^{h^*} + 2\alpha_{h^*,i^*} \geqslant f(x^\star)$, and thus $3LR\delta^{h^*} \geqslant f(x^\star) - f(x_{h^*,i^*})$ (by $\alpha_{h^*,i^*} = LR\delta^{h^*}$). Therefore, for any $\ell \in \{0,\dots,I_\varepsilon - 1\}$ (the case $\ell = 0$ is straightforward),

$$x_{h^*_\ell,i^*_\ell} \in \mathcal{X}_{3LR\delta^{h^*_\ell}} \,. \tag{4.8}$$

Now, let $k \in \{1,\dots,m_\varepsilon\}$ and $x_{h^*_\ell,i^*_\ell} \in \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]} \cap \mathcal{E}_\varepsilon$. By (4.8) and the fact that $x_{h^*_\ell,i^*_\ell} \in \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}$ is not ε_k-optimal, we have $3LR\delta^{h^*_\ell} > \varepsilon_k$. This and the definition of $h_{\varepsilon,k}$ entail

$$3LR\delta^{h_{\varepsilon,k}} > \varepsilon_k \,. \tag{4.9}$$

Also, let $x_{h,j}$ and $x_{h',j'}$ be two distinct elements of $\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]} \cap \mathcal{E}_\varepsilon$. By Assumption 4 and (4.9), we have $\|x_{h,j} - x_{h',j'}\| \geqslant \nu\delta^{\max\{h,h'\}} > \frac{\nu\varepsilon_k}{3LR}$. Therefore, and by definition of a packing number, we get that for all $k \in \{1,\dots,m_\varepsilon\}$, the cardinality $N_{\varepsilon,k}$ of $\mathcal{E}_\varepsilon \cap \mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}$ satisfies

$$N_{\varepsilon,k} \leqslant \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\nu\varepsilon_k}{3LR}\right) \leqslant \underbrace{\left(\mathbb{1}_{\frac{\nu}{3R}\geqslant 1} + \mathbb{1}_{\frac{\nu}{3R}<1}\left(1+\frac{6R}{\nu}\right)^n\right)}_{=:b} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) \tag{4.10}$$

from Lemma 4.11 in Appendix 4.B.

Now, let $x_{h^*_\ell,i^*_\ell} \in \mathcal{X}_\varepsilon \cap \mathcal{E}_\varepsilon$, with $\ell \in \{0,\dots,I_\varepsilon - 1\}$. If $\ell \geqslant 1$, we have, by definition of I_ε,

$$r_{h^*_\ell,i^*_\ell} + LR\delta^{h^*_\ell} + \alpha_{h^*_\ell,i^*_\ell} > \max_{s \leqslant T_\ell}\{r_s - \alpha_s\} + \varepsilon \geqslant r_{h^*_\ell,i^*_\ell} - \alpha_{h^*_\ell,i^*_\ell} + \varepsilon \,.$$

Again, replacing $\alpha_{h^*_\ell,i^*_\ell}$ with $LR\delta^{h^*_\ell}$, we get $3LR\delta^{h^*_\ell} > \varepsilon$, which is also true if $\ell = 0$. Therefore,

$$3LR\delta^{h_{\varepsilon,m_\varepsilon+1}} > \varepsilon \,. \tag{4.11}$$

Combining this inequality with Assumption 4, we get that $\|x_{h,j} - x_{h',j'}\| > \frac{\nu\varepsilon}{3LR}$ for any two distinct elements $x_{h,j}, x_{h',j'}$ of $\mathcal{X}_\varepsilon \cap \mathcal{E}_\varepsilon$. Therefore, and by definition of a packing number,

$$N_{\varepsilon,m_\varepsilon+1} \leqslant \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\nu\varepsilon}{3LR}\right) \leqslant b \cdot \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) \,, \tag{4.12}$$

by Lemma 4.11 again. Putting (4.4), (4.7), (4.10), (4.12) together and setting $a := Kb$, we get

$$\begin{aligned}
&\sigma(\text{c.MF-DOO}, E, \varepsilon) \\
&\quad \leqslant c(LR) + a\mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) c\left(LR\delta^{h_{\varepsilon,m_\varepsilon+1}+1}\right) + a\sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) c\left(LR\delta^{h_{\varepsilon,k}+1}\right) \\
&\quad \leqslant c(LR) + a\mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) c\left(\frac{\delta\varepsilon}{3}\right) + a\sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) c\left(\frac{\delta\varepsilon_k}{3}\right),
\end{aligned}$$

where we used (4.9), (4.11), and the fact that c is non-increasing. This concludes the proof. □

4.3 Lower bound

In this section, for any fixed L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$, we derive a lower bound on the worst-environment cost complexity $\sup_{E\in\mathcal{E}(f)} \sigma(A, E, \varepsilon)$ (see (4.1)) of any certified algorithm A. Our main result below, which depends on f through the key quantity $S_\beta(f, \varepsilon)$ defined in (4.2), generalizes (Bachoc et al., 2021, Theorem 2) from perfect evaluations of f to the multi-fidelity setting. We recall that $\varepsilon_0 = L\,\mathrm{diam}(\mathcal{X})$ and $m_\varepsilon = \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$.

Theorem 4.3. *Assume that $c : \mathbb{R}_+ \to \mathbb{R}_+$ is a non-increasing function and that $\mathcal{X} \subset \mathbb{R}^n$ is a compact and connected set. Then, for some constant $a_n > 0$ (e.g., $a_n = 1/65^n$), the cost complexity of any certified algorithm A satisfies, for any L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$ and any target optimization error $\varepsilon \in (0, \varepsilon_0/2)$,*

$$\sup_{E\in\mathcal{E}(f)} \sigma(A, E, \varepsilon) \geqslant \frac{a_n(1 - \mathrm{Lip}(f)/L)^n}{1 + m_\varepsilon} S_{16}(f, \varepsilon).$$

We make three comments before proving the theorem.

On the optimality of the bound First note that a_n depends exponentially on the dimension n. While removing such exponential dependence completely is challenging without stronger assumptions on f (if not impossible), the constant 65 was not optimized and could be improved. Besides, the quantity $(1 - \mathrm{Lip}(f)/L)^n$ vanishes as L approaches $\mathrm{Lip}(f)$. Importantly, the case $L = \mathrm{Lip}(f)$ is not really relevant in practice, because it scarcely happens that one knows exactly the best Lipschitz constant $\mathrm{Lip}(f)$ without knowing the function itself. In the more realistic case when one only knows a strict upper bound L on $\mathrm{Lip}(f)$, and under the mild assumption $\sup_{\alpha>0} c(\alpha)/c(2\alpha) < +\infty$ (which holds, e.g., if $c(\alpha)$ is polynomial in $1/\alpha$), Theorems 4.2 and 4.3 imply that c.MF-DOO is nearly optimal (among all certified algorithms) in terms of cost complexity, up to logarithmic and dimension-dependent multiplicative factors.

Earlier lower bounds Similarly to Section 4.2, Theorem 4.3 can be compared to (at least) two types of existing lower bounds. First, our lower bound generalizes that of (Bachoc et al., 2021, Theorem 2) (where f can be evaluated perfectly at the same cost as

coarse evaluations) to the multi-fidelity setting, where costs play a crucial role. Note that a study of the boundary case $L = \mathrm{Lip}(f)$ was provided by (Bachoc et al., 2021, Section 4), with different phenomena appearing in dimensions $n = 1$ or $n \geqslant 2$. Though out of the scope of this paper and with limited practical consequences, it would be interesting to investigate whether similar phenomena occur in our multi-fidelity setting.

A second type of lower bound (of a minimax form) was proved in (Fiegel et al., 2020, Theorem 1) for *non-certified* algorithms, under several assumptions on the cost function (more precisely, on a so-called *cost-to-bias* function) and a near-optimality dimension of f. Unlike the minimax approach, our lower bound is f-dependent. This is possible since we work with *certified* algorithms, whose data-driven certificates must be robust to yet unobserved values of f.

On more collaborative environments The lower bound of Theorem 4.3 holds for the worst case among all environments. However, the cost complexity can be improved for some specific environments. Indeed one could think of the following collaborative environment: when asked two times for an approximation of $f(x)$ with two accuracies α and α' at the same $x \in \mathcal{X}$, it first returns $f(x) - \alpha$ and then $f(x) + \alpha'$. Then even with $\alpha = \alpha' = \varepsilon_0$, the algorithm has an exact knowledge of $f(x)$ after only two queries at the same x. Against such an environment, we would thus be in the same setting as in Bachoc et al. (2021) (perfect evaluations of f) with only twice as many queries, and could therefore achieve a cost complexity of the order of $c(\varepsilon_0) \cdot \left(\mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) + \sum_{k=1}^{m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k, \varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right)\right)$. Since $c(\varepsilon_0)$ can be much smaller than $c(\varepsilon)$ in practice, this would greatly improve over the upper bound of Theorem 4.2, which (by Theorem 4.3) is nearly optimal when considering worst-case environments $E \in \mathcal{E}(f)$. In practice we could expect the environment to lie between the collaborative and worst-case extremes. The question of deriving environment-dependent lower and upper bounds is left for future work.

The proof of Theorem 4.3 is inspired from that of (Bachoc et al., 2021, Theorem 2) who addressed the case of perfect evaluations of f. Our generalization to the multi-fidelity setting however requires additional technicalities. Before the proof, we introduce several useful quantities and lemmas. Recall that $\mathcal{F}_L$ denotes the set of all L-Lipschitz functions from $\mathcal{X}$ to $\mathbb{R}$. We first define the quantity $\mathrm{err}_\tau(A, E)$ for any $\tau \geqslant 1$, as the best certificate ξ_τ that algorithm A could output given the sequence $(x_t, \alpha_t, y_t)_{t \leqslant \tau}$ and given x_τ^* (note that we consider all L-Lipschitz functions $g : \mathcal{X} \to \mathbb{R}$ that are compatible with the observations $(y_t)_{t \leqslant \tau}$):

$$\mathrm{err}_\tau(A, E) = \sup\Big\{\max(g) - g(x_\tau^*) : g \in \mathcal{F}_L \text{ and } \forall t \leqslant \tau, g(x_t) \in [y_t - \alpha_t, y_t + \alpha_t]\Big\}\,.$$

As can be seen from the next lemma, for any certified algorithm A, its certificate ξ_τ at any time $\tau \geqslant 1$ is bounded from below by $\mathrm{err}_\tau(A, E)$. The proof is postponed to Appendix 4.A.

Lemma 4.4. *Let $f : \mathcal{X} \to \mathbb{R}$ be an L-Lipschitz function, $E \in \mathcal{E}(f)$ be an environment and A be a certified algorithm with certificates $(\xi_t(E))_{t \geqslant 1}$ when run against E.*[8] *Then for all $\tau \in \mathbb{N}^*$, $\xi_\tau(E) \geqslant \mathrm{err}_\tau(A, E)$.*

[8] As mentioned in the introduction, ξ_t is a function of all values $y_1, \ldots, y_t$ observed so far. We stress the (implicit) dependency on E since it is key in the proof.

Denoting the set of all certified algorithms by $\mathcal{A}$, we can now define[9]

$$C_{\inf}(f,\varepsilon) = \inf\left\{C \in \mathbb{R} : \exists A \in \mathcal{A}, \forall E \in \mathcal{E}(f), \exists \tau \in \mathbb{N}^*, \sum_{t=1}^{\tau} c(\alpha_t(E)) \leqslant C \text{ and } \mathrm{err}_\tau(A,E) \leqslant \varepsilon\right\}$$

which represents the minimum cost that certified algorithms must incur to maximize f with an error certifiably below ε against any environment. This intuitive fact is formalized in the following lemma, which is proved in Appendix 4.A.

Lemma 4.5. *Let A be a certified algorithm, $f : \mathcal{X} \to \mathbb{R}$ be an L-Lipschitz function, and $\varepsilon \in (0, \varepsilon_0/2)$. Then, $\sup_{E\in\mathcal{E}(f)} \sigma(A,E,\varepsilon) \geqslant C_{\inf}(f,\varepsilon)$.*

Another intuitive result is that an algorithm cannot output a certificate $\xi_t \leqslant \varepsilon$ unless it has already requested some value of f with an evaluation accuracy $\alpha_t \leqslant \varepsilon$. This implies that the total cost needed to certify an error at level ε must be at least of $c(\varepsilon)$. This is stated formally below and proved in Appendix 4.A. Interestingly, this result would not hold if we worked with specific, possibly collaborative, environments (see a remark above).

Lemma 4.6. *Assume $\mathcal{X} \subset \mathbb{R}^,$ is compact and connected, and $c : \mathbb{R}_+ \to \mathbb{R}_+$ non-increasing. Then $C_{\inf}(f,\varepsilon) \geqslant c(\varepsilon)$ for any L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$ and any $\varepsilon \in (0, \varepsilon_0/2)$.*

We can now prove Theorem 4.3.

Proof of Theorem 4.3. We assume without loss of generality that $\mathrm{Lip}(f) < L$ and set $\Omega_f = \left(\frac{1-\mathrm{Lip}(f)/L}{65}\right)^n$. We want to show that

$$\sup_{E\in\mathcal{E}(f)} \sigma(A,E,\varepsilon) \geqslant \frac{\Omega_f}{1+m_\varepsilon} S_{16}(f,\varepsilon)\,.$$

Since $\sup_{E\in\mathcal{E}(f)} \sigma(A,E,\varepsilon) \geqslant C_{\inf}(f,\varepsilon)$ (by Lemma 4.5), it is sufficient to show that $C_{\inf}(f,\varepsilon) \geqslant \frac{\Omega_f}{1+m_\varepsilon} S_{16}(f,\varepsilon)$. We set $K = \frac{16L}{L-\mathrm{Lip}(f)}$ and note that $\Omega_f \leqslant \frac{1}{(1+4K)^n}$. We can distinguish between two cases:

First case: Assume first that $\frac{S_{16}(f,\varepsilon)}{1+m_\varepsilon} \leqslant (1+4K)^n c(\varepsilon)$. Then

$$\frac{\Omega_f}{1+m_\varepsilon} S_{16}(f,\varepsilon) \leqslant c(\varepsilon) \leqslant C_{\inf}(f,\varepsilon)\,,$$

where the last inequality follows from Lemma 4.6. In this case, the theorem is proved.

Second case: We now assume that $\frac{S_{16}(f,\varepsilon)}{1+m_\varepsilon} > (1+4K)^n c(\varepsilon)$. The idea is to upper bound the average of the $(1+m_\varepsilon)$ terms that define $S_{16}(f,\varepsilon)$ by the largest one.

Let $\widetilde{\varepsilon}$ be the scale with maximum contribution in (4.2) with $\beta = 16$, that is:

$$\widetilde{\varepsilon} = \begin{cases} \varepsilon & \text{if } \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) c(16\varepsilon) \geqslant \max_{1\leqslant k\leqslant m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) c(16\varepsilon_k) \\ \varepsilon_{k^*-1} & \text{otherwise, where } k^* \in \arg\max_{1\leqslant k\leqslant m_\varepsilon} \mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) c(16\varepsilon_k) \end{cases}$$

Since $\mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right) \leqslant \mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{2L}\right)$ and $\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right) \leqslant \mathcal{N}\left(\mathcal{X}_{\varepsilon_{k-1}}, \frac{\varepsilon_{k-1}}{2L}\right)$ for all $1 \leqslant k \leqslant m_\varepsilon$, and since c is non-increasing, we then have $S_{16}(f,\varepsilon) \leqslant (1+m_\varepsilon)\mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{\widetilde{\varepsilon}}{2L}\right) c(8\widetilde{\varepsilon})$.

[9] To see that the set of C's is never empty, take $A =$ c.MF-DOO and apply Theorem 4.2 and Lemma 4.4.

Then, using Lemma 4.11, the previous result, and the assumption of the second case, we have:

$$\begin{aligned}\mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right) c(8\widetilde{\varepsilon}) &\geqslant \left(\frac{1}{1+4K}\right)^n \mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{\widetilde{\varepsilon}}{2L}\right) c(8\widetilde{\varepsilon}) \\ &\geqslant \left(\frac{1}{1+4K}\right)^n \frac{S_{16}(f,\varepsilon)}{(1+m_\varepsilon)} > c(\varepsilon)\,. \end{aligned} \tag{4.13}$$

To prove our result, we assume for a moment that $C_{\inf}(f,\varepsilon) < \frac{\Omega_f}{1+m_\varepsilon} S_{16}(f,\varepsilon)$ and will show that it raises a contradiction. Combining with (4.13) and $\Omega_f \leqslant \frac{1}{(1+4K)^n}$, this indeed yields

$$C_{\inf}(f,\varepsilon) < \Omega_f (1+4K)^n \mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right) c(8\widetilde{\varepsilon}) \leqslant \mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right) c(8\widetilde{\varepsilon})\,.$$

Then, by definition of $C_{\inf}(f,\varepsilon)$, there exists an algorithm $A \in \mathcal{A}$ such that for all environments $E \in \mathcal{E}(f)$, there exists $\tau \in \mathbb{N}^*$ such that

$$\sum_{t=1}^{\tau} c(\alpha_t(E)) < \mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right) c(8\widetilde{\varepsilon}) \text{ and } \mathrm{err}_\tau(A,E) \leqslant \varepsilon\,. \tag{4.14}$$

We now consider the "noiseless" environment $E = (E_t)_{t\geqslant 1} \in \mathcal{E}(f)$ defined by $E_t(x,\alpha) = f(x)$ for all $t \geqslant 1$, $x \in \mathcal{X}$, and $\alpha > 0$. Let $\tau \in \mathbb{N}^*$ be such that (4.14) holds. Let $M = \mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right)$, and let $\{\widetilde{x}_1, \dots, \widetilde{x}_M\}$ be a $(K\widetilde{\varepsilon}/L)$-packing of $\mathcal{X}_{\widetilde{\varepsilon}}$. Note that the closed balls $B(\widetilde{x}_m, \frac{K\widetilde{\varepsilon}}{2L})$ with centers $\widetilde{x}_1, \dots, \widetilde{x}_M$ and radius $K\widetilde{\varepsilon}/2L$ are pairwise disjoint. Note also that $M \geqslant 2$ from $\mathcal{N}\left(\mathcal{X}_{\widetilde{\varepsilon}}, \frac{K\widetilde{\varepsilon}}{L}\right) > \frac{c(\varepsilon)}{c(8\widetilde{\varepsilon})} \geqslant 1$ by (4.13), $8\widetilde{\varepsilon} \geqslant \varepsilon$, and c being non-increasing ($c(8\widetilde{\varepsilon}) > 0$ by (4.13)).

For any $1 \leqslant m \leqslant M$, let c_m be the maximum cost spent at any round on the m-th ball, that is $c_m = \max\{c(\alpha_t) : t \in \mathcal{T}_m\}$ if the set $\mathcal{T}_m := \{t = 1, \dots, \tau : \|x_t(E) - \widetilde{x}_m\| \leqslant K\widetilde{\varepsilon}/2L\}$ is non-empty, and $c_m = 0$ otherwise. We know from (4.14) that the total cost up to round τ is smaller than $Mc(8\widetilde{\varepsilon})$. By the pigeonhole principle, there is at least one $m \leqslant M$ for which $c_m < c(8\widetilde{\varepsilon})$. Assume without loss of generality that this is true for $m = 1$. Then for any $t \in \mathcal{T}_1$ (if such t exists), the cost $c(\alpha_t(E))$ is smaller than $c(8\widetilde{\varepsilon})$. Therefore, either $\alpha_t(E) > 8\widetilde{\varepsilon}$ whenever the ball $B\left(\widetilde{x}_1, \frac{K\widetilde{\varepsilon}}{2L}\right)$ is visited (since c is non-increasing) or this ball is never visited.

We just showed that, on the ball $B(\widetilde{x}_1, \frac{K\widetilde{\varepsilon}}{2L})$, algorithm A never queried f with an evaluation accuracy $\alpha_t \leqslant 8\widetilde{\varepsilon}$. Next we show the following consequence: that the inequality $\mathrm{err}_\tau(A,E) \leqslant \varepsilon$ in (4.14) cannot be true, by exhibiting an L-Lipschitz function $g \in \mathcal{F}_L$ compatible with the observations $r_t = E_t(x_t(E), \alpha_t(E)) = f(x_t(E))$ and such that $\max(g) - g(x^*_\tau(E)) > \varepsilon$. This will raise a contradiction in (4.14) and conclude the proof. To that end, we consider the two functions $g = f \pm h_{\widetilde{\varepsilon}}$, with $h_{\widetilde{\varepsilon}} : \mathcal{X} \to \mathbb{R}$ defined by

$$h_{\widetilde{\varepsilon}}(x) = \max\left\{8\widetilde{\varepsilon} - 16\frac{L}{K}\|x - \widetilde{x}_1\|, 0\right\}\,. \tag{4.15}$$

First note that both $f - h_{\widetilde{\varepsilon}}$ and $f + h_{\widetilde{\varepsilon}}$ are L-Lipschitz, since $h_{\widetilde{\varepsilon}}$ is $(L - \mathrm{Lip}(f))$-Lipschitz (by $\frac{16L}{K} = L - \mathrm{Lip}(f)$). Moreover, since $h_{\widetilde{\varepsilon}}$ is supported on $\mathcal{X} \cap B(\widetilde{x}_1, K\widetilde{\varepsilon}/2L)$ and $\|h_{\widetilde{\varepsilon}}\|_\infty \leqslant$

$8\widetilde{\varepsilon} \leqslant \alpha_t(E)$ for all $t \in \mathcal{T}_1$, the two functions $f - h_{\widetilde{\varepsilon}}$ and $f + h_{\widetilde{\varepsilon}}$ belong by construction to the set

$$\mathcal{G} := \left\{ g \in \mathcal{F}_L : \forall t = 1, \dots, \tau,\ g(x_t(E)) \in \left[f(x_t(E)) - \alpha_t(E), f(x_t(E)) + \alpha_t(E) \right] \right\}.$$

We now show that $\max(g) - g(x^*_\tau(E)) > \varepsilon$ for $g = f - h_{\widetilde{\varepsilon}}$ or $g = f + h_{\widetilde{\varepsilon}}$, by distinguishing two subcases. If $x^*_\tau(E) \in B(\widetilde{x}_1, K\widetilde{\varepsilon}/4L)$, we perturb f "downwards" around $\widetilde{x}_1$ and consider $g = f - h_{\widetilde{\varepsilon}}$. In this case, since $h_{\widetilde{\varepsilon}}(x^*_\tau(E)) \geqslant 4\widetilde{\varepsilon}$ and $h_{\widetilde{\varepsilon}}(\widetilde{x}_2) = 0$, we have $\max(g) - g(x^*_\tau(E)) \geqslant f(\widetilde{x}_2) - h_{\widetilde{\varepsilon}}(\widetilde{x}_2) - (f(x^*_\tau(E)) - h_{\widetilde{\varepsilon}}(x^*_\tau(E))) \geqslant -\widetilde{\varepsilon} + 4\widetilde{\varepsilon} = 3\widetilde{\varepsilon}$.[10] In the other case, if $x^*_\tau(E) \notin B(\widetilde{x}_1, K\widetilde{\varepsilon}/4L)$, we consider $g = f + h_{\widetilde{\varepsilon}}$: since $h_{\widetilde{\varepsilon}}(x^*_\tau(E)) \leqslant 4\widetilde{\varepsilon}$ and $h_{\widetilde{\varepsilon}}(\widetilde{x}_1) = 8\widetilde{\varepsilon}$, we have $\max(g) - g(x^*_\tau(E)) \geqslant f(\widetilde{x}_1) + h_{\widetilde{\varepsilon}}(\widetilde{x}_1) - (f(x^*_\tau(E)) + h_{\widetilde{\varepsilon}}(x^*_\tau(E))) \geqslant -\widetilde{\varepsilon} + 8\widetilde{\varepsilon} - 4\widetilde{\varepsilon} = 3\widetilde{\varepsilon}$.
In both subcases above, we proved $\max(g) - g(x^*_\tau(E)) \geqslant 3\widetilde{\varepsilon} > \varepsilon$ for some $g \in \{f - h_{\widetilde{\varepsilon}}, f + h_{\widetilde{\varepsilon}}\} \subset \mathcal{G}$, which entails $\mathrm{err}_\tau(A, E) > \varepsilon$ (by definition of err_τ). This raises a contradiction in (4.14), so that we must have $C_{\inf}(f, \varepsilon) \geqslant \frac{\Omega_f}{1+m_\varepsilon} S_{16}(f, \varepsilon)$. This concludes the proof. □

4.4 Special case: noisy evaluations of f (a.k.a. stochastic setting)

Previously all the environments E that we considered were deterministic. We now assume that the algorithm receives noisy (stochastic and unbiased) evaluations of f, but that for all $t \in \mathbb{N}^*$ it can observe several independent noisy evaluations of $f(x_t)$ and decide the number m_t of them.

More formally, we consider the following variant of the online protocol described in Section 4.1.1. Let $(\zeta_{t,u})_{t,u \in \mathbb{N}^*}$ be a sequence of independent v-subGaussian random variables.[11] The $\zeta_{t,u}$'s are unknown, but the constant $v > 0$ is assumed to be known to the learner. At each round $t \in \mathbb{N}^*$, the algorithm A chooses a query point $x_t \in \mathcal{X}$, as well as a number $m_t \geqslant 1$ of noisy evaluations (instead of α_t). The algorithm incurs a cost equal to m_t. In return, the environment outputs a mini-batch of rewards $(r_{t,1}, \dots, r_{t,m_t})$ with m_t components (instead of a single inaccurate evaluation r_t), where $r_{t,u} = f(x_t) + \zeta_{t,u}$ for any $1 \leqslant u \leqslant m_t$. Then, just as before, A outputs a recommendation $x^*_t \in \mathcal{X}$ for the maximum of f, together with a (tentative) error certificate $\xi_t \geqslant 0$. In this setting, the goal is (with high probability) to maximize f with an error certifiably below ε, while minimizing the total number of evaluations of f.

To that end, we reduce the problem to the deterministic setting of Section 4.2. We consider c.MF-StoOO (Certified Multi-Fidelity Stochastic Optimistic Optimization), which is a mini-batch version of c.MF-DOO and whose pseudo-code is given in Algorithm 12 below. In the sequel, we use the same identification between nodes and rounds as before (see Footnote 5).

The intuition behind c.MF-StoOO is the following: for a mini-batch of size m_t, the average $r_t := \frac{1}{m_t} \sum_{u=1}^{m_t} r_{t,u}$ is an unbiased estimate of $f(x_t)$ with a variance bounded by v/m_t. To be in a special case of Section 4.2, we make the standard deviation $\sqrt{v/m_t}$ comparable to the evaluation accuracy α_t that c.MF-DOO would request, by choosing $m_t \approx v/\alpha_t^2$ and then applying Theorem 4.2 with a cost $c(\alpha) \approx v/\alpha^2$. More precisely, we

[10] We used the fact that $f(\widetilde{x}_m) - f(x^*_\tau(E)) \geqslant f(\widetilde{x}_m) - \max(f) \geqslant -\widetilde{\varepsilon}$ for all $1 \leqslant m \leqslant M$ (since $\widetilde{x}_m \in \mathcal{X}_{\widetilde{\varepsilon}}$).

[11] A real-valued random variable X is v-subGaussian if $\mathbb{E}[\exp(\lambda X)] \leqslant \exp(\lambda^2 v/2)$ for all $\lambda \in \mathbb{R}$.

use a careful weighted union bound on the nodes of the hierarchical partitioning tree. For some desired risk level $\gamma \in (0,1)$, writing (h_t, i_t) for the node evaluated at time $t \geqslant 1$ (Line 9 if $t \geqslant 2$), we take

$$m_t = \left\lceil \frac{2v}{\alpha_t^2} \ln\left(\frac{2}{\gamma_{h_t,i_t}}\right) \right\rceil \text{, with } \gamma_{h,i} = \frac{\gamma}{(h+1)(h+2)K^h} \text{ for } h \in \mathbb{N} \text{ and } i \in \{0,\dots,K^h-1\}. \tag{4.16}$$

Note that the weights sum up to $\sum_{h=0}^{+\infty}\sum_{i=0}^{K^h-1}\gamma_{h,i} = \gamma$.

Algorithm 12 c.MF-StoOO (Certified Multi-Fidelity Stochastic Optimistic Optimization)

Inputs: $\mathcal{X}$, K, $(X_{h,i})_{h\in\mathbb{N},i\in\{0,\dots,K^h-1\}}$, $(x_{h,i})_{h\in\mathbb{N},i\in\{1,\dots,K^h-1\}}$, δ, R, L, v, and γ
Initialization Let $t \leftarrow 1$, $(h_1,i_1) \leftarrow (0,0)$, and $\mathcal{L}_1 \leftarrow \{(0,0)\}$

1: Pick the first query point $x_1 \leftarrow x_{0,0}$, accuracy $\alpha_1 \leftarrow LR$, prior value $\gamma_{0,0} \leftarrow \gamma/2$, and evaluation number $m_1 \leftarrow \left\lceil \frac{2v}{(LR)^2} \ln\left(\frac{4}{\gamma}\right) \right\rceil$
2: Observe the noisy evaluations $(r_{1,u})_{1\leqslant u\leqslant m_1} = (f(x_1)+\zeta_{1,u})_{1\leqslant u\leqslant m_1}$
3: Compute $r_1 = \frac{1}{m_1}\sum_{u=1}^{m_1} y_{1,u}$
4: Output the recommendation $x_1^* \leftarrow x_1$ and certificate $\xi_1 \leftarrow LR$
5: Pick the first node $(h^*,i^*) \leftarrow (0,0)$
6: **for** *iteration* $= 1,2,\dots$ **do**
7: **for all** child (h^*+1,j) of (h^*,i^*) **do**
8: **if** $\mathcal{X}_{h^*+1,j} \cap \mathcal{X} \neq \varnothing$ **then**
9: Let $t \leftarrow t+1$, $(h_t,i_t) \leftarrow (h^*+1,j)$ and $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{(h_t,i_t)\}$
10: Pick the next query point $x_t \leftarrow x_{h_t,i_t}$ and accuracy $\alpha_t \leftarrow LR\delta^{h_t}$
11: Compute the prior value $\gamma_{h_t,i_t} = \gamma/((h_t+1)(h_t+2)K^{h_t})$
12: Pick the number of evaluations $m_t \leftarrow \left\lceil \frac{2v}{\alpha_t^2} \ln\left(\frac{2}{\gamma_{h_t,i_t}}\right) \right\rceil$
13: Observe the noisy evaluations $(r_{t,u})_{1\leqslant u\leqslant m_t} = (f(x_t)+\zeta_{t,u})_{1\leqslant u\leqslant m_t}$
14: Compute $r_t = \frac{1}{m_t}\sum_{u=1}^{m_t} y_{t,u}$
15: Output the recommendation $x_t^* = x_{\widetilde{t}}$, with $\widetilde{t} \in \arg\max_{1\leqslant s\leqslant t}\{r_s - \alpha_s\}$
16: Output the certificate $\xi_t = r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} - (r_{\widetilde{t}} - \alpha_{\widetilde{t}})$
17: **end if**
18: **end for**
19: Remove (h^*,i^*) from $\mathcal{L}_t$
20: Let $(h^*,i^*) \in \arg\max_{(h,i)\in\mathcal{L}_t}\{r_{h,i} + LR\delta^h + \alpha_{h,i}\}$
21: Update the last certificate $\xi_t = r_{h^*,i^*} + LR\delta^{h^*} + \alpha_{h^*,i^*} - (r_{\widetilde{t}} - \alpha_{\widetilde{t}})$
22: **end for**

Define the stopping time

$$\tau(f,\varepsilon) := \inf\{t \geqslant 1 : \xi_t \leqslant \varepsilon\}\,.$$

We now bound the total number $\sum_{t=1}^{\tau(f,\varepsilon)} m_t$ of evaluations of f that c.MF-StoOO requests before certifying an ε-maximizer of f. The next high-probability bound (Proposition 4.7)

is in terms of the quantity $S_\beta(f,\varepsilon)$ defined in (4.2) with the cost function $c = c_\gamma$ given by

$$c_\gamma(\alpha) = \left\lceil \frac{2v}{\alpha^2} \ln\left(\frac{2(h(\alpha)+1)(h(\alpha)+2)K^{h(\alpha)}}{\gamma} \right) \right\rceil, \quad \text{where} \quad h(\alpha) = \frac{\ln(LR/\alpha)}{\ln(1/\delta)}\,. \tag{4.17}$$

We recall that $\varepsilon_0 = L\,\mathrm{diam}(\mathcal{X})$, $m_\varepsilon = \lceil \log_2(\varepsilon_0/\varepsilon) \rceil$, $\varepsilon_{m_\varepsilon} = \varepsilon$ and $\varepsilon_k = \varepsilon_0 2^{-k}$ for $1 \leqslant k \leqslant m_\varepsilon - 1$.

Proposition 4.7. *Suppose that $\mathcal{X} \subset \mathbb{R}^m$ is compact, that Assumptions 3 and 4 hold, and denote by $a > 0$ the same constant as in Theorem 4.2. Then, in the stochastic setting described above, for any known constants $L, v > 0$ and $\gamma \in (0,1)$, for any L-Lipschitz function $f : \mathcal{X} \to \mathbb{R}$ with maximizer $x^\star \in \mathcal{X}$, and any $\varepsilon \in (0, \varepsilon_0)$, c.MF-StoOO (Algorithm 12) satisfies*

$$\mathbb{P}\left(\left[\forall t \geqslant 1, f(x^\star) - f(x_t^*) \leqslant \xi_t \right] \text{ and } \sum_{t=1}^{\tau(f,\varepsilon)} m_t \leqslant a S_{\frac{\delta}{3}}(f,\varepsilon) + \left\lceil \frac{2v}{(LR)^2} \ln\left(\frac{4}{\gamma}\right) \right\rceil \right) \geqslant 1 - \gamma\,,$$

where the probability is taken over the noise sequence $(\zeta_{t,u})_{t,u\in\mathbb{N}^*}$, and where $S_{\delta/3}(f,\varepsilon)$ is the quantity defined in (4.2) with $\beta = \delta/3$ and $c = c_\gamma$ (see (4.17) above).

We first comment on the result. Proposition 4.7 implies that, with high probability, the ξ_t's are valid certificates and the total number of evaluations of f that c.MF-StoOO requests before certifying an ε-maximizer of f is bounded roughly by (combining (4.2) with (4.17), and omitting log factors and some dimension-dependent constants)

$$\frac{v\mathcal{N}\left(\mathcal{X}_\varepsilon, \frac{\varepsilon}{L}\right)}{\varepsilon^2} + \sum_{k=1}^{m_\varepsilon} \frac{v\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}, \frac{\varepsilon_k}{L}\right)}{\varepsilon_k^2} \;\approx\; v L^n \int_{\mathcal{X}} \frac{\mathrm{d}x}{(f(x^\star) - f(x) + \varepsilon)^{n+2}}\,,$$

where the sum-integral approximation (which omits multiplicative constants) holds under the following mild geometric condition on $\mathcal{X}$: there exist constants $r_0 > \mathrm{diam}(\mathcal{X})/2$, $\rho \in (0,1]$ such that for all $x \in \mathcal{X}$ and $r \in (0, r_0)$, $\mathrm{vol}(B(x,r) \cap \mathcal{X}) \geqslant \rho\,\mathrm{vol}(B(x,r))$. This condition roughly states that $\mathcal{X}$ has a non-negligible volume locally everywhere (e.g., we can take $r_0 = 1$ and $\rho = 2^{-n}$ if $\mathcal{X} = [0,1]^n$ and $\|\cdot\|$ is the sup norm). The proof of this sum-integral approximation follows essentially from (Bachoc et al., 2021, Theorem 1), with a direct extension to non-increasing costs, and is show in Appendix 4.D.

Therefore, a consequence of Proposition 4.7 is that, under a mild condition on $\mathcal{X}$, the cost complexity in the stochastic setting is roughly proportional to $\int_{\mathcal{X}} dx/(f(x^\star) - f(x) + \varepsilon)^{n+2}$, as conjectured by Bachoc et al. (2021).

To prove Proposition 4.7, we use the following classical lemma, which helps reduce the stochastic setting with mini-batches to the deterministic setting with inaccurate evaluations. (We will later combine this lemma with $r_t - f(x_t) = \frac{1}{m_t}\sum_{u=1}^{m_t}(r_{t,u} - f(x_t)) = \frac{1}{m_t}\sum_{u=1}^{m_t}\zeta_{t,u}$.)

Lemma 4.8. *Let $(\zeta_{t,u})_{t,u\in\mathbb{N}^*}$ be a sequence of independent v-subGaussian random variables for some $v > 0$. Let $\gamma \in (0,1)$, and let $(\alpha_t)_{t\geqslant 1}$ and $(h_t, i_t)_{t\geqslant 1}$ be two predictable[12] sequences*

[12] That is, we assume α_t, h_t and i_t to be measurable w.r.t. the subsequence $(\zeta_{s,u})_{1\leqslant s\leqslant t-1, u\in\mathbb{N}^*}$.

such that, almost surely, $\alpha_t > 0$, $h_t \in \mathbb{N}$ and $i_t \in \{0, \dots, K^{h_t} - 1\}$ for all $t \geqslant 1$, and $t \geqslant 1 \mapsto (h_t, i_t)$ being injective. Then, for m_t and $\gamma_{h,i}$ defined as in (4.16), we have

$$\mathbb{P}\left(\forall t \in \mathbb{N}^*, \left|\frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u}\right| < \alpha_t\right) \geqslant 1 - \gamma\,.$$

Proof. Let $\mathcal{F}_t$ denote the σ-field generated by the random variables $\zeta_{s,u}$, $s \in \{1, \dots, t\}$, $u \in \mathbb{N}^*$. (By convention, $\mathcal{F}_0$ is the trivial σ-field.) Let $t \geqslant 1$. Since α_t and m_t are $\mathcal{F}_{t-1}$ measurable, and the $\zeta_{t,u}$, $u \in \mathbb{N}^*$, are independent and v-subGaussian conditionally on $\mathcal{F}_{t-1}$,

$$\mathbb{P}\left(\left|\frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u}\right| \geqslant \alpha_t\right) = \mathbb{E}\left[\mathbb{P}\left(\left|\frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u}\right| \geqslant \alpha_t \middle| \mathcal{F}_{t-1}\right)\right] \leqslant \mathbb{E}\left[2e^{-\frac{m_t\alpha_t^2}{2v}}\right] \leqslant \mathbb{E}\left[\gamma_{h_t,i_t}\right]\,.$$

By a union bound, this yields

$$\mathbb{P}\left(\exists t \in \mathbb{N}^*, \left|\frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u}\right| \geqslant \alpha_t\right) \leqslant \sum_{t=1}^{+\infty} \mathbb{P}\left(\left|\frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u}\right| \geqslant \alpha_t\right) \leqslant \mathbb{E}\left[\sum_{t=1}^{+\infty} \gamma_{h_t,i_t}\right] \leqslant \gamma\,,$$

where the last inequality follows by injectivity of $t \geqslant 1 \mapsto (h_t, i_t)$ and $\sum_{h=0}^{+\infty}\sum_{i=0}^{K^h-1}\gamma_{h,i} = \gamma$. Taking the complementary event concludes the proof. □

Proof of Proposition 4.7. We now explain how to treat the problem as a special case of Section 4.2 (deterministic yet inaccurate evaluations), with the cost function c_γ defined in (4.17) and a well-chosen random environment.

First note that the assumptions of Lemma 4.8 are met, so that with high probability the y_t's are α_t-close to $f(x_t)$ simultaneously for all $t \geqslant 1$. More formally, they can be seen as generated by a random environment $(E_t^\omega)_{t\geqslant 1}$ defined as follows. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the probability space on which the random variables $(\zeta_{t,u})_{t,u}$ are defined. For any fixed element $\omega \in \Omega$ and any $t \geqslant 1$, we define the function $E_t^\omega : \mathcal{X} \times \mathbb{R}_+^* \to \mathbb{R}$ by

$$E_t^\omega(x, \alpha) = \begin{cases} f(x) + \frac{1}{m_t(\omega)}\sum_{u=1}^{m_t(\omega)} \zeta_{t,u}(\omega) & \text{if } \left|\frac{1}{m_t(\omega)}\sum_{u=1}^{m_t(\omega)} \zeta_{t,u}(\omega)\right| \leqslant \alpha \\ f(x) & \textit{otherwise.} \end{cases}$$

Note that the environment $(E_t^\omega)_{t\geqslant 1}$ lies in $\mathcal{E}(f)$ for each $\omega \in \Omega$. Now, denoting by $\widetilde{\Omega} = \left\{\omega \in \Omega : \forall t \in \mathbb{N}^*, \left|\frac{1}{m_t(\omega)}\sum_{u=1}^{m_t(\omega)} \zeta_{t,u}(\omega)\right| < \alpha_t\right\}$ the event considered in Lemma 4.8, we can see that, for any $\omega \in \widetilde{\Omega}$,[13] the value $E_t^\omega(x_t, \alpha_t) = f(x_t) + \frac{1}{m_t}\sum_{u=1}^{m_t} \zeta_{t,u} = \frac{1}{m_t}\sum_{u=1}^{m_t}\big(f(x_t) + \zeta_{t,u}\big)$ coincides with $r_t = \frac{1}{m_t}\sum_{u=1}^{m_t} r_{t,u}$ for all $t \geqslant 1$. We can thus apply Lemma 4.1: for any $\omega \in \widetilde{\Omega}$, we have $f(x^\star) - f(x_t^*) \leqslant \xi_t$ for all $t \geqslant 1$ (ξ_t is a valid certificate).

Furthermore, a call to $E_t^\omega(x_t, \alpha_t)$ requires $m_t = \left\lceil \frac{2v}{\alpha_t^2} \ln\left(\frac{2}{\gamma_{h_t,i_t}}\right)\right\rceil = c_\gamma(\alpha_t)$ noisy evaluations of f, with c_γ defined in (4.17). Putting everything together, for any $\omega \in \widetilde{\Omega}$, the behavior of c.MF-StoOO coincides with the behavior of c.MF-DOO against the environment $E_\omega \in \mathcal{E}(f)$. In particular the total cost $\sum_{t=1}^{\tau(f,\varepsilon)} m_t = \sum_{t=1}^{\tau(f,\varepsilon)} c_\gamma(\alpha_t)$ coincides with the

[13] For the sake of readability, we drop some dependencies in ω.

cost complexity of c.MF-DOO against E_ω, with the non-increasing cost function c_γ. We can thus use Theorem 4.2: for any $\omega \in \widetilde{\Omega}$,

$$\sum_{t=1}^{\tau(f,\varepsilon)} m_t = \sigma(\text{c.MF-DOO}, E^\omega, \varepsilon) \leqslant a S_{\frac{\delta}{3}}(f,\varepsilon) + c_\gamma(LR) \,.$$

Recalling that $\mathbb{P}(\widetilde{\Omega}) \geqslant 1-\gamma$ concludes the proof. □

Acknowledgements

The authors would like to thank François Bachoc for insightful feedback. This work has benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French "Investing for the Future–PIA3" program under the Grant agreement ANR-19-P3IA-0004. The authors gratefully acknowledge the support of the DEEL project.[14]

4.A Missing proofs in Section 4.3

Proof of Lemma 4.4. Let $\tau \in \mathbb{N}^*$. For the sake of clarity, we explicitly write the dependencies of the iterates $x_t(E), \alpha_t(E), r_t(E), x_t^*(E), \xi_t(E)$ w.r.t. the environment $E \in \mathcal{E}(f)$. Recall that $\mathcal{F}_L$ denotes the set of all L-Lipschitz functions from $\mathcal{X}$ to $\mathbb{R}$. Let $g \in \mathcal{F}_L$ be such that

$$\forall t \leqslant \tau, g(x_t(E)) \in [r_t(E) - \alpha_t(E), r_t(E) + \alpha_t(E)] \,. \tag{4.18}$$

Then there exists an environment $E^g \in \mathcal{E}(g)$ whose interactions with algorithm A yield the same decisions x_t, α_t, x_t^*, ξ_t and observations r_t as those generated with environment E, up to time $t = \tau$. More formally, we define $E^g = (E_t^g)_{t \geqslant 1}$ by $E_t^g(x,\alpha) = y_t(E)$ if $t \leqslant \tau$, $x = x_t(E)$ and $\alpha = \alpha_t(E)$, but $E_t^g(x,\alpha) = g(x)$ otherwise. Note that $E^g \in \mathcal{E}(g)$ by (4.18). First note that $x_1(E^g) = x_1(E)$ and $\alpha_1(E^g) = \alpha_1(E)$ since both terms are independent of the environment. From our definition of E^g, this implies that the approximation $r_1(E^g) = E_1^g(x_1(E^g), \alpha_1(E^g))$ returned by E^g is equal to $r_1(E)$. Because the observation r_1 that A receives is the same as before, A outputs the same values for x_1^*, ξ_1, x_2 and α_2, which again implies that $r_2(E^g) = r_2(E)$. By a simple induction argument, we then have that, for all $t = 1, \dots, \tau$, $x_t(E^g) = x_t(E)$, $\alpha_t(E^g) = \alpha_t(E)$, $r_t(E^g) = r_t(E)$, $x_t^*(E^g) = x_t^*(E)$ and $\xi_t(E^g) = \xi_t(E)$. In particular,

$$\xi_\tau(E) = \xi_\tau(E_g) \geqslant \max(g) - g(x_\tau^*(E_g)) = \max(g) - g(x_\tau^*(E)) \,,$$

where the inequality follows from the definition of a certificate. Since the above lower bound on $\xi_\tau(E)$ is true for all $g \in \mathcal{F}_L$ satisfying condition (4.18), it is also true for their supremum, which proves that $\xi_\tau(E) \geqslant \mathrm{err}_\tau(A,E)$. □

Proof of Lemma 4.5. Let A be a certified algorithm. We can assume without loss of generality that $\sup_{E \in \mathcal{E}(f)} \sigma(A,E,\varepsilon) < +\infty$. Let $C > \sup_{E \in \mathcal{E}(f)} \sigma(A,E,\varepsilon)$. Then, for any

[14] https://www.deel.ai/

$E \in \mathcal{E}(f)$, by definition of $\sigma(A,E,\varepsilon)$, there exists $\tau \in \mathbb{N}^*$ such that $\sum_{t=1}^{\tau} c(\alpha_t(E)) \leqslant C$ and $\xi_\tau(E) \leqslant \varepsilon$. By Lemma 4.4 and since A is a certified algorithm, this entails

$$\sum_{t=1}^{\tau} c(\alpha_t(E)) \leqslant C \text{ and } \mathrm{err}_\tau(A,E) \leqslant \varepsilon.$$

By definition of $C_{\inf}(f,\varepsilon)$, this immediately yields $C_{\inf}(f,\varepsilon) \leqslant C$. We conclude the proof by letting C go to $\sup_{E\in\mathcal{E}(f)} \sigma(A,E,\varepsilon)$. □

In order to prove Lemma 4.6, we first need the following intuitive lemma.

Lemma 4.9. *Let f be any L-Lipschitz function, $\tau \in \mathbb{N}^*$, and let E^* be the "noiseless" environment $E^* = ((x,\alpha) \mapsto f(x))_{t\geqslant 1} \in \mathcal{E}(f)$. Then, for any certified algorithm A, the best possible certificate $\mathrm{err}_\tau(A,E^*)$ against E^* is bounded from below by $\min\{\min_{t\leqslant\tau} \alpha_t(E^*), \varepsilon_0/2\}$.*

Proof. Within this proof, we only work with the environment E^*, so we skip all dependencies of $x_t, \alpha_t, r_t, x_t^*, \xi_t$ on E^*. We set $\widetilde{\varepsilon} = \min\{\min_{t\leqslant\tau} \alpha_t, \varepsilon_0/2\}$ and define $g : \mathcal{X} \to \mathbb{R}$ by $g(x) = \min\{f(x) + \widetilde{\varepsilon}, f(x_\tau^*) - \widetilde{\varepsilon} + L\|x - x_\tau^*\|\}$. We will now show that g is compatible with the observations $(y_t)_{t\leqslant\tau}$ and the accuracies $(\alpha_t)_{t\leqslant\tau}$, and that $\max(g) - g(x_\tau^*) \geqslant \widetilde{\varepsilon}$.

Since g is the minimum of two L-Lipschitz functions, it is also L-Lipschitz. Moreover, for any $t \leqslant \tau$, on the one hand, $g(x_t) \leqslant f(x_t) + \widetilde{\varepsilon} \leqslant r_t + \alpha_t$, by definition of g and $\widetilde{\varepsilon}$, and the fact that $r_t = f(x_t)$ (recall that we work with the "noiseless" environment E^*).

On the other hand, by L-Lipschitz continuity of f, for all $x \in \mathcal{X}$, $L\|x - x_\tau^*\| + f(x_\tau^*) - \widetilde{\varepsilon} \geqslant f(x) - \widetilde{\varepsilon}$. This implies that for all $x \in \mathcal{X}$, $g(x) \geqslant \min\{f(x) + \widetilde{\varepsilon}, f(x) - \widetilde{\varepsilon}\} = f(x) - \widetilde{\varepsilon}$. In particular, for all $t \leqslant \tau$,

$$g(x_t) \geqslant f(x_t) - \widetilde{\varepsilon} \geqslant r_t - \alpha_t\,.$$

To sum up, g is an L-Lipschitz function such that $|g(x_t) - r_t| \leqslant \alpha_t$ for all $1 \leqslant t \leqslant \tau$: the algorithm A cannot make the difference between the functions f and g.

Now, let us bound $\max(g) - g(x_\tau^*)$ from below to derive a lower bound on $\mathrm{err}_\tau(A,E^*)$. First, $g(x_\tau^*) = \min\{f(x_\tau^*) + \widetilde{\varepsilon}, f(x_\tau^*) - \widetilde{\varepsilon}\} = f(x_\tau^*) - \widetilde{\varepsilon}$. Second, let $v \in \mathbb{R}^n$ be such that $\|v\|_2 = 1$ and $x_\tau^* + \frac{\widetilde{\varepsilon} v}{L} \in \mathcal{X}$. Such a v exists by Lemma 4.10 and the facts that $\widetilde{\varepsilon} \leqslant \varepsilon_0/2 = L \cdot \mathrm{diam}(\mathcal{X})/2$ and $\mathcal{X}$ is compact and connected. Then,

$$f\left(x_\tau^* + \frac{\widetilde{\varepsilon} v}{L}\right) + \widetilde{\varepsilon} \geqslant f(x_\tau^*) - L\left\|\frac{\widetilde{\varepsilon} v}{L}\right\| + \widetilde{\varepsilon} = f(x_\tau^*)$$

and

$$f(x_\tau^*) - \widetilde{\varepsilon} + L\left\|x_\tau^* + \frac{\widetilde{\varepsilon} v}{L} - x_\tau^*\right\| = f(x_\tau^*) - \widetilde{\varepsilon} + L\left\|\frac{\widetilde{\varepsilon} v}{L}\right\| = f(x_\tau^*)\,.$$

By definition of g, this entails

$$f(x_\tau^*) \leqslant g\left(x_\tau^* + \frac{\widetilde{\varepsilon} v}{L}\right) \leqslant \max(g)$$

Putting everything together, we get

$$\mathrm{err}_\tau(A,E^*) \geqslant \max(g) - g(x_\tau^*) \geqslant f(x_\tau^*) - (f(x_\tau^*) - \widetilde{\varepsilon}) = \widetilde{\varepsilon} = \min\left\{\min_{t\leqslant\tau} \alpha_t, \varepsilon_0/2\right\},$$

which concludes the proof. □

Proof of Lemma 4.6. Recall the definition of $C_{\inf}(f,\varepsilon)$, and let $C \in \mathbb{R}$ and $A' \in \mathcal{A}$ be such that for all $E \in \mathcal{E}(f)$, there exists $\tau \in \mathbb{N}^*$ such that $\sum_{t=1}^{\tau} c(\alpha_t(E)) \leqslant C$ and $\mathrm{err}_\tau(A', E) \leqslant \varepsilon$. In particular, for the "noiseless" environment $E^* = ((x,\alpha) \mapsto f(x))_{t\geqslant 1}$, there exists τ such that $\sum_{t=1}^{\tau} c(\alpha_t(E^*)) \leqslant C$ and $\mathrm{err}_\tau(A, E^*) \leqslant \varepsilon$. Using Lemma 4.9, we get that $\min\{\min_{t\leqslant\tau} \alpha_t(E^*), \varepsilon_0/2\} \leqslant \mathrm{err}_\tau(A, E^*) \leqslant \varepsilon$, which implies $\min_{t\leqslant\tau} \alpha_t(E^*) \leqslant \varepsilon$ (since $\varepsilon_0/2 > \varepsilon$ by assumption). Because c is non-negative and non-increasing, we have:

$$C \geqslant \sum_{t=1}^{\tau} c(\alpha_t(E^*)) \geqslant c\left(\min_{t\leqslant\tau} \alpha_t(E^*)\right) \geqslant c(\varepsilon)\,.$$

By definition of $C_{\inf}(f,\varepsilon)$, this entails that $C_{\inf}(f,\varepsilon) \geqslant c(\varepsilon)$. □

4.B Useful lemmas

We recall two rather classical lemmas, and provide proofs for the convenience of the reader.

Lemma 4.10. *Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact connected set with diameter ρ. Then for any $x \in \mathcal{X}$ and $\varepsilon \leqslant \frac{\rho}{2}$, there exists $v \in \mathbb{R}^n$ with $\|v\| = 1$ such that $x + \varepsilon v \in \mathcal{X}$.*

Proof. Let $x \in \mathcal{X}$ and $0 < \varepsilon \leqslant \frac{\rho}{2}$ (the result is straightforward if $\varepsilon = 0$). Since ρ is the diameter of $\mathcal{X}$ and the latter is compact, there exist $y, z \in \mathcal{X}$ such that $\|y - z\| = \rho$.
Let us show by contradiction that there exists a point $x' \in \mathcal{X}$ for which $\|x - x'\| \geqslant \varepsilon$. Assume for a moment that for all $x' \in \mathcal{X}$, $\|x - x'\| < \varepsilon$. In that case, we would have $\|y - z\| \leqslant \|y - x\| + \|x - z\| < 2\varepsilon \leqslant \rho$, which is in contradiction with $\|y - z\| = \rho$.

Now that we know that it exists, let $x' \in \mathcal{X}$ be such that $\|x - x'\| \geqslant \varepsilon$. Because $\mathcal{X}$ is connected, there exists a continuous path $\gamma : t \in [0,1] \mapsto \gamma(t) \in \mathcal{X}$ such that $\gamma(0) = x$ and $\gamma(1) = x'$. Let g be the function $g : t \in [0,1] \mapsto \|\gamma(t) - x\| \in \mathbb{R}$. g is a continuous function from $[0,1]$ to $\mathbb{R}$, with $g(0) = 0$ and $g(1) \geqslant \varepsilon > 0$, so according to the intermediate value theorem, there exists $t^* \in [0,1]$ such that $g(t^*) = \varepsilon$. Taking $x'' = \gamma(t^*)$ and $v = \frac{x'' - x}{\|x - x''\|}$ solves the problem, because $x + \varepsilon v = x'' \in \mathcal{X}$. □

Lemma 4.11. *For any bounded set $E \subset \mathbb{R}^n$, and all $0 < \rho_1 < \rho_2$, we have*

$$\mathcal{N}(E, \rho_1) \leqslant \left(1 + 2\frac{\rho_2}{\rho_1}\right)^n \mathcal{N}(E, \rho_2).$$

The above lemma is well known and can be found, e.g., in (Bachoc et al., 2021, Appendix A) with a slightly weaker statement. We recall the proof for the convenience of the reader. For any $x \in \mathbb{R}^n$ and $\rho > 0$, we set $B(x,\rho) = \{u \in \mathbb{R}^n : \|u - x\| \leqslant \rho\}$.

Proof. Fix any bounded set $E \subset \mathbb{R}^n$ and $0 < \rho_1 < \rho_2$. Consider an ρ_1-packing $F = \{x_1, \ldots, x_{N_1}\}$ of E, with cardinality $N_1 := \mathcal{N}(E, \rho_1)$.
Let $F_0 = F$. We define a sequence $F_0, F_1, \ldots, F_{k_{\mathrm{end}}-1}$ of subsets of F by induction, as follows. For $k \geqslant 1$ let $\widehat{x}_k$ be any element of F_{k-1}, and define $B_k = F_{k-1} \cap B(\widehat{x}_k, \rho_2)$ and $F_k = F_{k-1} \backslash B_k$. Repeating this procedure, we get an index $k_{\mathrm{end}} \leqslant N_1$ such that $F_{k_{\mathrm{end}}-1}$ is non-empty while $F_{k_{\mathrm{end}}}$ is empty.
Then, the set $\{\widehat{x}_1, \ldots, \widehat{x}_{k_{\mathrm{end}}}\}$ is an ρ_2-packing of E, so that $k_{\mathrm{end}} \leqslant \mathcal{N}(E, \rho_2)$. Let us now upper bound N_1 using k_{end}. By construction, the union of the B_k's contains F, so for all

$i \leqslant N_1$, there exists $k \leqslant k_{\mathrm{end}}$ such that $x_i \in B_k$, and thus $B(x_i, \rho_1/2) \subset B(\widehat{x}_k, \rho_2 + \rho_1/2)$. Therefore,

$$\bigcup_{1 \leqslant i \leqslant N_1} B(x_i, \rho_1/2) \subset \bigcup_{1 \leqslant k \leqslant k_{\mathrm{end}}} B(\widehat{x}_k, \rho_2 + \rho_1/2)\,.$$

Moreover, the N_1 balls $B(x_i, \rho_1/2)$ are pairwise disjoint, because F is an ρ_1-packing of E. By a volumetric argument, we thus get that $(\rho_1/2)^d N_1 \leqslant (\rho_2 + \rho_1/2)^d k_{\mathrm{end}} \leqslant (\rho_2 + \rho_1/2)^d \mathcal{N}(E, \rho_2)$. Rearranging terms concludes the proof. □

4.C The special case of constant costs

In this short section we focus on the case of a constant cost $c(\alpha) = 1$. We formalize and prove the following very intuitive fact: when more accurate evaluations come at no additional cost, choosing the best accuracy available is always optimal. To avoid boundary effects, we slightly extend the setting by allowing the learner to choose α_t identically equal to zero.

Lemma 4.12. *Let $f : \mathcal{X} \to \mathbb{R}$ be an L-Lipschitz function, and assume $c : \mathbb{R}_+ \to \mathbb{R}_+$ is the constant function given by $c(\alpha) = 1$ for all $\alpha \geqslant 0$. Then,*

- $\sigma(A, E, \varepsilon) = \inf\{t \in \mathbb{N}^* : \xi_t(E) \leqslant \varepsilon\}$;
- *the smallest cost complexity (against the worst environment) is achieved by certified algorithms that choose $\alpha_t = 0$ for all $t \in \mathbb{N}^*$.*

Proof of Lemma 4.12. Let f be an L-Lipschitz function, and $\varepsilon > 0$. Since $c(\alpha) = 1$ for all $\alpha \geqslant 0$, the definition of $\sigma(A, E, \varepsilon)$ directly yields $\sigma(A, E, \varepsilon) = \inf\{t \in \mathbb{N}^* : \xi_t(E) \leqslant \varepsilon\}$.

Recall that $\mathcal{A}$ denotes the set of all certified algorithms, and that $\mathcal{E}(f)$ is the set of all environments associated with the function f. As discussed in the introduction, α_t depends on E only via the inaccurate approximations $r_1, \dots, r_{t-1}$ of f. We write $\alpha_t(r_1, \dots, r_{t-1})$ instead of α_t to clarify this dependency when needed. We denote by $\mathcal{A}_0$ the set of all certified algorithms such that $\alpha_t(r_1, \dots, r_{t-1}) = 0$ for all $t \in \mathbb{N}^*$ against all possible realizations of $r_1, \dots, r_{t-1}$.

Formally, what we want to prove is

$$\inf_{A' \in \mathcal{A}_0} \sup_{E \in \mathcal{E}(f)} \sigma(A', E, \varepsilon) \leqslant \inf_{A \in \mathcal{A}} \sup_{E \in \mathcal{E}(f)} \sigma(A, E, \varepsilon)\,.$$

Let A be an algorithm in $\mathcal{A}$, and let $E^* \in \mathcal{E}(f)$ be the environment for which $E^*_t(x, \alpha) = f(x)$ for all $t \in \mathbb{N}^*$, all $x \in \mathcal{X}$ and all $\alpha \geqslant 0$. Note that

$$\sup_{E \in \mathcal{E}(f)} \sigma(A, E, \varepsilon) \geqslant \sigma(A, E^*, \varepsilon) \tag{4.19}$$

Now, let $\widetilde{A}$ be the same algorithm as A, with one difference: whatever the situation and the environment E, for all $t \in \mathbb{N}$, $\widetilde{\alpha}_t(r_1, \dots, r_{t-1}) = 0$. Then $\widetilde{A} \in \mathcal{A}_0$. Moreover, because of the particularity of E^*, A and $\widetilde{A}$ will behave the same way against E^*: the sequences of query points $(x_t)_{t \in \mathbb{N}^*}$, recommendations $(x^*_t)_{t \in \mathbb{N}^*}$ and certificates $(\xi_t)_{t \in \mathbb{N}^*}$ are the same

for A and for $\widetilde{A}$. From this, we get that $\sigma(A,E^*,\varepsilon)=\sigma(\widetilde{A},E^*,\varepsilon)$. Combining with (4.19) yields

$$\inf_{A\in\mathcal{A}}\sup_{E\in\mathcal{E}(f)}\sigma(A,E,\varepsilon)\geqslant\inf_{A\in\mathcal{A}}\sigma(A,E^*,\varepsilon)\geqslant\inf_{\widetilde{A}\in\mathcal{A}_0}\sigma(\widetilde{A},E^*,\varepsilon) \tag{4.20}$$

Let us now show that

$$\inf_{A'\in\mathcal{A}_0}\sigma(A',E^*,\varepsilon)=\inf_{A'\in\mathcal{A}_0}\sup_{E\in\mathcal{E}(f)}\sigma(A',E,\varepsilon), \tag{4.21}$$

which, combined with (4.20), will conclude the proof.

To see why (4.21) holds, let A' be any certified algorithm with $\alpha_t=0$ for all $t\in\mathbb{N}^*$, whatever the past observations $r_1,\dots,r_{t-1}$, and let E, and E' be two environments in $\mathcal{E}(f)$. Then, for all $x\in\mathcal{X}$, $E_t(x,0)=E'_t(x,0)=f(x)$, because environments should satisfy $|E_t(x,\alpha)-f(x)|\leqslant\alpha$ for all $x\in\mathcal{X}$ and $\alpha\geqslant 0$. Therefore, the behavior of A' against any $E\in\mathcal{E}(f)$ is the same as against E^* which proves (4.21). □

4.D On the characterization of $S_\beta(f,\varepsilon)$

This section is greatly inspired by (Bachoc et al., 2021, Section 3). The only difference in their results and ours is the apparition of the cost function c, which changes slightly the results, but not the details of the proofs.

In all this section, for any radius $\rho>0$, we denote by v_ρ the radius of any ball with radius ρ. Note that $v_\rho=\rho^n v_1$. For a radius ρ and a set $\mathcal{X}'\subset\mathcal{X}$, we denote by $\mathcal{M}(\mathcal{X}',\rho)$ the r-covering number of $\mathcal{X}'$. It consists of the minimal number m of points $x'_1,\dots,x'_m\in\mathcal{X}'$ that covers $\mathcal{X}'$ with closed ball of radius ρ, that is such that $\mathcal{X}'\subset\bigcup_{i=1}^m\overline{B}(x_i,\rho)$. In order to characterize $S_\beta(f,\varepsilon)$, we need the following assumption on the volume of any ball in the set $\mathcal{X}$:

Assumption 5. There exist two constants $\rho_0>0$ and $\gamma\in(0,1]$ such that for any $x\in\mathcal{X}$ and all $\rho\in(0,\rho_0)$, $\mathrm{vol}(B_\rho(x)\cap\mathcal{X})\geqslant\gamma v_\rho$.

Lemma 4.13. *Let f be any L-Lipschitz function. For any $x\in\mathcal{X}$, let us denote by Δ_x the gap $\max_{x'\in\mathcal{X}}f(x')-f(x)$. If Assumption 5 holds with $\rho_0>\varepsilon_0/2L$ and some $\gamma\in(0,1]$, then there exist two constants $0<c_n<C_n$ such that for all $\varepsilon\in(0,\varepsilon_0)$,*

$$c_n\int_{x\in\mathcal{X}}\frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}\,\mathrm{d}x\leqslant S_\beta(f,\varepsilon)\leqslant C_n\int_{x\in\mathcal{X}}\frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}\,\mathrm{d}x$$

More precisely, we have $c_n={}^{L^n}\!/_{v_1}$ and $C_n={}^{(32L)^n}\!/_{\gamma v_1}$.

Proof. Fix any $\varepsilon\in(0,\varepsilon_0)$ and recall that $m_\varepsilon:=\lceil\log_2({}^{\varepsilon_0}\!/_{\varepsilon})\rceil$, $\varepsilon_{m_\varepsilon}:=\varepsilon$ and for all $1\leqslant k\leqslant m_\varepsilon-1$, $\varepsilon_k:=\varepsilon_0 2^{-k}$. We begin by proving the first inequality:

$$\begin{aligned}
\int_{x\in\mathcal{X}}\frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}\,\mathrm{d}x&\leqslant\frac{\mathrm{vol}(\mathcal{X}_\varepsilon)c(\beta\varepsilon)}{\varepsilon^n}+\sum_{k=1}^{m_\varepsilon}\frac{c(\beta(\varepsilon_k+\varepsilon))}{(\varepsilon_k+\varepsilon)^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})\\
&\leqslant\frac{\mathcal{M}(\mathcal{X}_\varepsilon,{}^{\varepsilon}\!/_{L})}{\varepsilon^n}v_1\left(\frac{\varepsilon}{L}\right)^n+\sum_{k=1}^{m_\varepsilon}\frac{c(\beta\varepsilon_k)}{\varepsilon_k^n}\mathcal{M}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]},{}^{\varepsilon_k}\!/_{L}\right)v_1\left(\frac{\varepsilon_k}{L}\right)^n\\
&\leqslant\frac{v_1}{L^n}\left(\mathcal{N}(\mathcal{X}_\varepsilon,{}^{\varepsilon}\!/_{L})\,c(\beta\varepsilon)+\sum_{k=1}^{m_\varepsilon}c(\beta\varepsilon_k)\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]},{}^{\varepsilon_k}\!/_{L}\right)\right).
\end{aligned}$$

The first inequality comes by bounding the decreasing function $x \mapsto \frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}$ by its infimum on each partition, and the second by a upper bounding the volume of a set by the volume of its smallest ε_k/L-covering. The last one comes from the fact that covering numbers are always smaller than packing numbers (Wainwright, 2019, Lemma 5.5, with permuted notation of $\mathcal{M}$ and $\mathcal{N}$).

For the second inequality, we have (where B_1 is the open unity ball):

$$\begin{aligned}
\int_{x\in\mathcal{X}} \frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}\,\mathrm{d}x &\geqslant \frac{\mathrm{vol}(\mathcal{X}_\varepsilon)c(\beta(\varepsilon+\varepsilon))}{(\varepsilon+\varepsilon)^n} + \sum_{k=1}^{m_\varepsilon} \frac{c(\beta(\varepsilon_{k-1}+\varepsilon))}{(\varepsilon_{k-1}+\varepsilon)^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}) \\
&\geqslant \frac{1}{2^n}\left(\frac{\mathrm{vol}(\mathcal{X}_\varepsilon)c(2\beta\varepsilon)}{\varepsilon^n} + \sum_{k=1}^{m_\varepsilon} \frac{c(2\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})\right) \\
&\geqslant \frac{1}{2\times 4^n}\left(\frac{\mathrm{vol}\left(\mathcal{X}_{2\varepsilon}\right)c(4\beta\varepsilon)}{\varepsilon^n} + \sum_{k=1}^{m_\varepsilon} \frac{c(4\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}\left(\mathcal{X}_{(\frac{\varepsilon_k}{2},2\varepsilon_{k-1}]}\right)\right) \\
&\geqslant \frac{\gamma}{2}\left(\frac{\mathcal{N}\left(\mathcal{X}_\varepsilon,\varepsilon/L\right)\mathrm{vol}\left(\frac{\varepsilon}{2L}B_1\right)c(4\beta\varepsilon)}{(4\varepsilon)^n} + \sum_{k=1}^{m_\varepsilon} \frac{\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]},\varepsilon_k/L\right)\mathrm{vol}\left(\frac{\varepsilon_k}{2L}B_1\right)c(4\beta\varepsilon_k)}{(4\varepsilon_{k-1})^n}\right) \\
&\geqslant \frac{\gamma}{2}v_{1/16L}\left(\mathcal{N}\left(\mathcal{X}_\varepsilon,\varepsilon/L\right)c(4\beta\varepsilon) + \sum_{k=1}^{m_\varepsilon}\mathcal{N}\left(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]},\varepsilon_k/L\right)c(4\beta\varepsilon_{k-1})\right) \\
&\geqslant \frac{\gamma}{2}v_{1/16L}S_{4\beta}(f,\varepsilon)\,.
\end{aligned}$$

Here, we bounded Δ_x by its supremum on the partition thanks to the decrease of $x \mapsto \frac{c(\beta(\Delta_x+\varepsilon))}{(\Delta_x+\varepsilon)^n}$ for the first inequality, we used $\varepsilon \leqslant \varepsilon_{k-1}$ and the decrease of the same function for the second inequality, Lemma 4.14 for the third one, and properties of packing number for the next ones. □

Lemma 4.14. *Let f be any L-Lipschitz functions and fix $\varepsilon > 0$. Then the following inequality holds:*

$$\frac{\mathrm{vol}\left(\mathcal{X}_{2\varepsilon}\right)c(\beta\varepsilon)}{\varepsilon^n} + \sum_{k=1}^{m}\mathrm{vol}\left(\mathcal{X}_{(\frac{\varepsilon_k}{2},2\varepsilon_{k-1}]}\right)\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n} \leqslant 2\times 2^n\left(\frac{\mathrm{vol}(\mathcal{X}_\varepsilon)}{\varepsilon^n}c\left(\frac{\beta\varepsilon}{2}\right) + \sum_{k=1}^{m}\frac{\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})}{\varepsilon_{k-1}^n}c\left(\frac{\beta\varepsilon_{k-1}}{2}\right)\right)$$

Proof. To avoid clutter, we denote m_ε simply by m. Then,

$$\begin{aligned}
&\frac{\mathrm{vol}\left(\mathcal{X}_{2\varepsilon}\right)c(\beta\varepsilon)}{\varepsilon^n} + \sum_{k=1}^{m}\mathrm{vol}\left(\mathcal{X}_{(\frac{\varepsilon_k}{2},2\varepsilon_{k-1}]}\right)\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n} \\
&\leqslant \frac{c(\beta\varepsilon)}{\varepsilon^n}\left(\mathrm{vol}(\mathcal{X}_\varepsilon) + \mathrm{vol}(\mathcal{X}_{(\varepsilon_m,\varepsilon_{m-1}]})\right) + \frac{c(\beta\varepsilon_{m-1})}{\varepsilon_{m-1}^n}\left(\mathrm{vol}(\mathcal{X}_\varepsilon) + \mathrm{vol}(\mathcal{X}_{(\varepsilon_m,\varepsilon_{m-1}]}) + \mathrm{vol}(\mathcal{X}_{(\varepsilon_{m-1},\varepsilon_{m-2}]})\right) \\
&+ \sum_{k=1}^{m-1}\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\left(\mathrm{vol}(\mathcal{X}_{(\varepsilon_{k+1},\varepsilon_k]}) + \mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}) + \mathrm{vol}(\mathcal{X}_{(\varepsilon_{k-1},\varepsilon_{k-2}]})\right)
\end{aligned}$$

$$
\begin{aligned}
&\leqslant 2\frac{c(\beta\varepsilon)}{\varepsilon^n}\mathrm{vol}(\mathcal{X}_\varepsilon) + 2^n\frac{\mathrm{vol}(\mathcal{X}_{(\varepsilon_m,\varepsilon_{m-1}]})}{\varepsilon_{m-1}^n}c\left(\frac{\beta\varepsilon_{m-1}}{2}\right)\\
&+\sum_{k=1}^{m-1}\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_{k+1},\varepsilon_k]}) + \sum_{k=1}^{m}\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}) + \sum_{k=1}^{m}\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_{k-1},\varepsilon_{k-2}]})\\
&\leqslant 2\frac{c(\beta\varepsilon)}{\varepsilon^n}\mathrm{vol}(\mathcal{X}_\varepsilon)\\
&+\sum_{k=2}^{m}\frac{c(2\beta\varepsilon_{k-1})}{2^n\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]}) + \sum_{k=1}^{m}\frac{c(\beta\varepsilon_{k-1})}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_k) + \sum_{k=1}^{m}c\left(\frac{\beta\varepsilon_{k-1}}{2}\right)\frac{2^n}{\varepsilon_{k-1}^n}\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})\\
&\leqslant \left(\frac{1}{2^n}+1+2^n\right)\left(\frac{c(\beta\varepsilon)}{\varepsilon^n}\mathrm{vol}(\mathcal{X}_\varepsilon) + \sum_{k=1}^{m}c\left(\frac{\beta\varepsilon_{k-1}}{2}\right)\frac{\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})}{\varepsilon_{k-1}^n}\right)\\
&\leqslant 2^{nx+1}\left(c\left(\frac{\beta\varepsilon}{2}\right)\frac{\mathrm{vol}(\mathcal{X}_\varepsilon)}{\varepsilon^n} + \sum_{k=1}^{m}c\left(\frac{\beta\varepsilon_{k-1}}{2}\right)\frac{\mathrm{vol}(\mathcal{X}_{(\varepsilon_k,\varepsilon_{k-1}]})}{\varepsilon_{k-1}^n}\right).
\end{aligned}
$$

We applied several times the definition of the ε_k's and their decrease with k. Moreover, we used the fact that $\frac{1}{2^n}+1+2^n \leqslant 2^{n+1}$ in the last inequality.

□

Chapter 5

Adaptive Approximation of Monotone Functions

We study the classical problem of approximating a non-decreasing function $f : \mathcal{X} \to \mathcal{Y}$ in $L^p(\mu)$ norm by sequentially querying its values, for known compact real intervals $\mathcal{X}$, $\mathcal{Y}$ and a known probability measure μ on $\mathcal{X}$. For any function f we characterize the minimum number of evaluations of f that algorithms need to guarantee an approximation $\widehat{f}$ with an $L^p(\mu)$ error below ε after stopping. Unlike worst-case results that hold uniformly over all f, our complexity measure is dependent on each specific function f. To address this problem, we introduce GreedyBox, a generalization of an algorithm originally proposed by Novak (1992) for numerical integration. We prove that GreedyBox achieves an optimal sample complexity for any function f, up to logarithmic factors. Additionally, we uncover results regarding piecewise-smooth functions. Perhaps as expected, the $L^p(\mu)$ error of GreedyBox decreases much faster for piecewise-C^2 functions than predicted by the algorithm (without any knowledge on the smoothness of f). A simple modification even achieves optimal minimax approximation rates for such functions, which we compute explicitly. In particular, our findings highlight multiple performance gaps between adaptive and non-adaptive algorithms, smooth and piecewise-smooth functions, as well as monotone or non-monotone functions. Finally, we provide numerical experiments to support our theoretical results.

This chapter is based on a joint work with Sébastien Gerchinovitz and Pierre Gaillard.

Contents

5.1 Introduction 115
5.1.1 Contributions and outline of the paper 117
5.1.2 Important definitions and notation 118
5.2 Lower bound 120
5.3 Upper bound 121

5.3.1 Algorithm and main result . . . 122
5.3.2 Proof of Theorem 5.3 . . . 124
5.4 Improvement for special cases . . . 125
5.4.1 Side problem: integral estimation . . . 125
Nearly optimal performance for integral estimation . . . 125
Improvement in expectation with randomization . . . 126
5.4.2 An intriguing result for piecewise-regular functions . . . 129
5.5 Numerical experiments . . . 131
5.6 Conclusion and future works . . . 135
5.A Elementary properties of box-covers . . . 137
5.A.1 A general upper bound (proof of Lemma 5.1) . . . 137
5.A.2 Technical lemmas on box-covers . . . 137
5.A.3 Relationship between $\mathcal{N}_p(f,\varepsilon)$ and $\mathcal{N}'_p(f,\varepsilon)$ . . . 138
5.B Omitted proofs . . . 140
5.B.1 Proof of Lemma 5.4 . . . 140
5.B.2 Proof of Lemma 5.5 . . . 142
5.B.3 Proof of Lemma 5.6 . . . 144
5.B.4 Proof of Theorem 5.10 . . . 145
5.B.5 Proof of Theorem 5.11 . . . 146
5.B.6 Proof of Proposition 5.12 . . . 149
5.B.7 Proof of Proposition 5.13 . . . 151
5.B.8 Technical lemmas for regular functions . . . 153
5.C Numerical experiments: the L^2-norm Case . . . 154

5.1 Introduction

Let $\mathcal{X},\mathcal{Y}$ be any non-empty compact intervals in $\mathbb{R}$. The problem we consider in this paper is the following. Given any non-decreasing function $f:\mathcal{X}\to\mathcal{Y}$ that is initially unknown but that a learner can sequentially evaluate at points $x_1,x_2,\ldots\in\mathcal{X}$ of their choice, how to best estimate f with as few evaluations of f as possible? We will study the $L^p(\mu)$ error as a performance criterion, for some known integer $p\geqslant 1$ and some known probability measure μ on $\mathcal{X}$. More precisely, we will study algorithms that can guarantee an $L^p(\mu)$ error observably below ε after a finite number of evaluations, and will characterize the minimum number of such evaluations to reach this goal, for any non-decreasing function f. Even though this problem is a classical one, finding the best f-dependent sample complexity and an algorithm that achieves it is still an open question.

To make things more formal, we first describe how the learner interacts with the unknown function f.

Online protocol. Given an accuracy level $\varepsilon > 0$, the learner first chooses a point $x_1 \in \mathcal{X}$, then observes $f(x_1) \in \mathcal{Y}$, then chooses $x_2 \in \mathcal{X}$, then observes $f(x_2) \in \mathcal{Y}$, etc. At each round $t \geqslant 2$, the point $x_t \in \mathcal{X}$ is chosen as a measurable function of the whole history $h_{t-1} := (x_1, f(x_1), \dots, x_{t-1}, f(x_{t-1}))$. The process ends after a finite number $\tau_\varepsilon \geqslant 1$ of rounds whose value may be determined during the observation process (τ_ε is a stopping time).[1] Finally, after observing the whole history h_{τ_ε}, the learner outputs a function $\widehat{f}_{\tau_\varepsilon} : \mathcal{X} \to \mathcal{Y}$ as a candidate for estimating f. We will call *algorithm*[2] any procedure that, given $\varepsilon > 0$ and f, returns a tuple $(\tau_\varepsilon, (x_t)_{1 \leqslant t \leqslant \tau_\varepsilon}, \widehat{f}_{\tau_\varepsilon})$ in $\mathbb{N}^* \times \mathcal{X}^{\tau_\varepsilon} \times (\mathcal{X} \to \mathcal{Y})$ satisfying the above online protocol. We only consider deterministic algorithms, except for the integral estimation problem (Section 5.4.1) for which randomized algorithms achieve better rates in expectation.

Learning goal: small number of evaluations with guaranteed $L^p(\mu)$ error. Let $p \geqslant 1$ be any positive integer and μ be any probability measure on $\mathcal{X}$. The performance of the learner will be evaluated by its $L^p(\mu)$ *error* defined by

$$\left\| \widehat{f}_{\tau_\varepsilon} - f \right\|_p := \left(\int_{\mathcal{X}} |\widehat{f}_{\tau_\varepsilon}(x) - f(x)|^p \, \mathrm{d}\mu(x) \right)^{1/p}. \tag{5.1}$$

In all the sequel, an accuracy level $\varepsilon > 0$ will be initially given to the learner, who will be required to guarantee that $\|\widehat{f}_{\tau_\varepsilon} - f\|_p \leqslant \varepsilon$ after stopping, for any (initially unknown) non-decreasing function $f : \mathcal{X} \to \mathcal{Y}$. Given this constraint, the goal of the learner is to make as few evaluations of f as possible, that is, to minimize the stopping time τ_ε. We will also refer to τ_ε as the *sample complexity* of the algorithm.

Main intuitions and informal presentation of the results. Before detailing our results in the next sections, we describe the main intuitions in the special case where $p = 1$ and μ is the Lebesgue measure on $\mathcal{X} = \mathcal{Y} = [0,1]$. The ideas are introduced informally and will be made more precise later.

Imagine that we have already evaluated f at some points $x_1, \dots, x_t \in [0,1]$. Since we know that f is non-decreasing and bounded between 0 and 1, we can deduce that the graph of f is contained inside the $t+1$ adjacent rectangles (or *boxes*) shown in Figure 5.1.[3] Therefore, estimating f with any function $\widehat{f}_t$ whose graph also lies in these $t+1$ adjacent boxes will guarantee an L^1 error $\|\widehat{f}_t - f\|_1$ of at most the total area ξ_t of these boxes. We can even achieve $\|\widehat{f}_t - f\|_1 \leqslant \xi_t/2$ by estimating f with a piecewise-constant function (on each box $B_j = [c_{j-1}, c_j] \times [y_j^-, y_j^+]$, choose $\widehat{f}_t(x) = (y_j^- + y_j^+)/2$) or with a piecewise-affine function (choose $\widehat{f}_t$ that linearly interpolates all the observed points $(c_j, f(c_j))$).

Now let $\varepsilon > 0$, and suppose that we want to guarantee an L^1 error below ε as quickly as possible. Given the above comment, it seems that an ideal choice of the sequence $x_1, x_2, \dots$

[1] This means that, for any integer $t \geqslant 1$, whether the inequality $\tau_\varepsilon \leqslant t$ is true or not is fully known after observing h_t (in a measurable way).

[2] Throughout the paper our definition of algorithms refers in fact to *adaptive algorithms* that adjust their sequence of points $x_1, \dots, x_t$ to the function f to be approximated based on previous observations. The latter should be contrasted with *non-adaptive algorithms*, for which the sequence of points $(x_t)_{t \geqslant 1}$ is fixed for all functions f's.

[3] Two boxes are degenerate (and thus not visible) on Figure 5.1, since GreedyBox evaluates f at the endpoints 0 and 1.

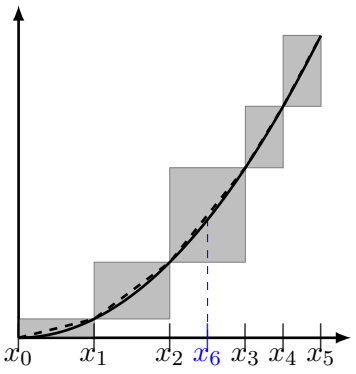


Figure 5.1: An illustrative example of the problem on the square function. After 5 iterations, the output of GreedyBox is represented by the gray boxes. The estimated function is represented by the solid line and its approximation by the dashed line. The next evaluation point x_6 divides the box with the largest area in half.

is such that the total area ξ_t of the $t+1$ adjacent boxes at round t falls below 2ε for the smallest value of t possible. We derive lower and upper bounds that support this intuition:

- Lower bound: if after stopping (at time τ_ε) we want to guarantee that $\|\widehat{f}_{\tau_\varepsilon} - f\|_1 \leqslant \varepsilon$ whatever f, then $\tau_\varepsilon + 1$ must be larger than or equal to the minimum number (denoted by $\mathcal{N}_1(f, 2\varepsilon)$) of adjacent boxes that contain the graph of f and whose total area is at most of 2ε (see Theorem 5.2).
- A nearly optimal greedy algorithm: of course an optimal choice of $x_1, x_2, \ldots$ is impractical (it would require the full knowledge of the function f). However, a natural algorithm is to choose at each round t the next point x_{t+1} in the middle of the box with maximum area, so as to greedily reduce the total area of the boxes. See Figure 5.1 for an illustration. This algorithm, which we call GreedyBox, was suggested in a similar form by Novak (1992). One of our main contributions is to show that the stopping time τ_ε of GreedyBox is always at most of the order of the lower bound $\mathcal{N}_1(f, 2\varepsilon)$ up to a logarithmic factor in $1/\varepsilon$ (see Theorem 5.3).

Both the lower and upper bounds are proved in a more general setting, in $L^p(\mu)$ norm, for any integer $p \geqslant 1$ and any probability measure μ on $\mathcal{X}$.

5.1.1 Contributions and outline of the paper

Our main contribution is to characterize the optimal sample complexity of algorithms with guaranteed $L^p(\mu)$ error after stopping (see after Equation (5.1)), for any non-decreasing function f, any $p \geqslant 1$ and any probability measure μ. More precisely:

- In Section 5.2 we prove a general f-dependent lower bound that applies to any algorithm with guaranteed $L^p(\mu)$ error after stopping (see Theorem 5.2).
- In Section 5.3 we study GreedyBox (Algorithm 13) and show that its sample complexity matches our lower bound up to logarithmic factors (see Theorem 5.3). An important practical feature of GreedyBox is that, at each iteration t, it provides a certificate that upper bounds its error and stops as soon as this certificate falls below ε.

All the results are written in the case where $\mathcal{X} = \mathcal{Y} = [0,1]$ for convenience, but all of them can be rescaled to any non-empty compact intervals $\mathcal{X}$ and $\mathcal{Y}$ of $\mathbb{R}$.[4]

In Section 5.4 we study consequences (with improved rates) for two specific subproblems:

- *Integral approximation.* For this problem, we show that the deterministic version of GreedyBox (Algorithm 13) is also optimal up to logarithmic factors. However, drawing inspiration from Novak (1992), we introduce a randomized version in Section 5.4.1 that improves the accuracy by a factor of $t^{-1/2}$ in expectation after t iterations (Theorem 5.10).
- *Worst-case function approximation under a smoothness assumption.* In the worst case, the upper bound of Theorem 5.3 is of the order of ε^{-1} for monotone functions. It is well known that for smooth functions, better rates can be achieved using improved quadrature formulas (e.g., Davis and Rabinowitz (2007)). For example, C^2 functions can be ε-approximated in any $L^p(\mu)$ norm after roughly $\varepsilon^{-1/2}$ evaluations (also discussed in Appendix 5.B.8). In Section 5.4.2, we provide a minimax lower bound showing that $\Omega\big(\varepsilon^{-1+(\frac{1-\alpha}{1+p})_+}\big)$ function evaluations are necessary for any algorithm seeking to approximate a piecewise-affine function with $\varepsilon^{-\alpha}$ singularities (see Proposition 5.12). We establish that GreedyBox in fact achieves this rate for piecewise-C^2 functions when $\alpha \geqslant 1/2$, but is suboptimal in the regime $0 \leqslant \alpha < 1/2$ (Theorem 5.11 and Proposition 5.13). Lastly, we propose a simple modification of GreedyBox that optimally addresses both regimes. These results highlight three significant differences for the L^p-approximation problem:
  - between monotone piecewise-C^k functions with two or more discontinuities, for which a better rate than $\varepsilon^{-1/2}$ is not achievable, and C^k functions (with no singularities) which can be approximated at a rate of $\mathcal{O}(\varepsilon^{-1/k})$;
  - between general piecewise-C^2 functions and monotone piecewise-C^2 functions, with a minimax rate respectively of at least $\Omega(\varepsilon^{-p})$ and at most $o(\varepsilon^{-1})$ for $\alpha < 1$;
  - between non-adaptive algorithms, which need $\Omega(\varepsilon^{-1})$ function evaluations, and adaptive ones, which only require $o(\varepsilon^{-1})$ for $\alpha < 1$.

Finally, in Section 5.5, we provide numerical experiments that compare GreedyBox to the trapezoidal method on several functions f. Our simulations validate the rates anticipated by our analysis and demonstrate the superiority of our approach compared to the uniform trapezoidal rule in approximating monotone piecewise-smooth functions within the L^p norm.

5.1.2 Important definitions and notation

We now introduce several definitions and notations that will be useful to present our results more formally. In all the sequel, we work with $\mathcal{X} = \mathcal{Y} = [0,1]$, some fixed integer $p \geqslant 1$ and a probability measure μ on $[0,1]$.

Standard notation. We denote by $\mathbb{N}^*$ the set of integers greater than or equal to 1. For any $x \in \mathbb{R}$, we denote the floor and ceiling functions at x by $\lfloor x \rfloor := \max\{k \in \mathbb{Z} : k \leqslant x\}$

[4] The case where $\mathcal{X}$ is not closed can be addressed similarly via a simple extension argument and by replacing the values of f at the endpoints of $\mathcal{X}$ by the endpoints of $\mathcal{Y}$.

and $\lceil x \rceil := \min\{k \in \mathbb{Z} : k \geqslant x\}$ respectively. For any measurable function $g : [0,1] \to \mathbb{R}$, we denote the $L^p(\mu)$-norm of g by $\|g\|_p := \left(\int_{[0,1]} |g(x)|^p \, \mathrm{d}\mu(x)\right)^{1/p}$.

Box, width, generalized area. We call *box* any subset $B = [x^-, x^+] \times [y^-, y^+] \subseteq [0,1]^2$ with $x^- < x^+$ and $y^- \leqslant y^+$. Its *width* is given by

$$\mathrm{width}(B) := \mu\big((x^-, x^+)\big)$$

and its *generalized area* is given by

$$\mathcal{A}_p(B) := \Big((y^+ - y^-)^p \, \mu\big((x^-, x^+)\big)\Big)^{1/p} .$$

Note that the above definitions consider open intervals (x^-, x^+). The generalized area corresponds to the usual notion of area when $p = 1$ and μ is the Lebesgue measure.

Adjacent boxes. We denote by $\mathcal{B}$ the set of all boxes. We say that a finite sequence of $t \geqslant 1$ boxes $B_1, \ldots, B_t \in \mathcal{B}$ are *adjacent* if and only if they are of the form $B_j = [c_{j-1}, c_j] \times [y_j^-, y_j^+]$ for all $j = 1, \ldots, t$, for some sequence $0 = c_0 < c_1 < \ldots < c_{t-1} < c_t = 1$.

Box-cover and box-covering number of a function. Let $f : [0,1] \to [0,1]$ be any non-decreasing function. We call *box-cover of f* any sequence $B_1, \ldots, B_t \in \mathcal{B}$ of adjacent boxes that contains the graph of f except maybe at the boxes' endpoints, i.e., writing $B_j = [c_{j-1}, c_j] \times [y_j^-, y_j^+]$ as above, such that $\big\{(x, f(x)) : x \in [0,1] \setminus \{c_0, \ldots, c_t\}\big\} \subseteq \cup_{i=1}^t B_i$. Furthermore, given $\varepsilon > 0$, we define two complexity quantities:

(i) $\mathcal{N}_p(f, \varepsilon)$ denotes the minimum cardinality t of a box-cover $B_1, \ldots, B_t$ of f whose generalized areas satisfy $\left(\sum_{i=1}^t \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon$. We call this quantity the *box-covering number of f at scale ε*.

(ii) $\mathcal{N}'_p(f, \varepsilon)$ denotes the minimum cardinality t of a box-cover $B_1, \ldots, B_t$ of f with generalized areas $\mathcal{A}_p(B_i) \leqslant \varepsilon$ for all $i = 1, \ldots, t$.

All our main results will be expressed in terms of $\mathcal{N}_p(f, \varepsilon)$; the other quantity $\mathcal{N}'_p(f, \varepsilon)$ will however be useful in the proofs. The connections between the two are described in Appendix 5.A.3.

We now reinterpret the condition $\left(\sum_{i=1}^t \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon$ appearing in (i) in an equivalent way. When $p = 1$, this condition corresponds to the total area of the boxes being bounded by ε (as mentioned earlier in the introduction). For general $p \geqslant 1$, an equivalent and useful formulation is the following. Denote by $B_j = [c_{j-1}, c_j] \times [y_j^-, y_j^+]$, $j = 1, \ldots, t$, the boxes of the cover, and by $f^-(x) := \sum_{j=1}^t y_j^- \mathbb{1}_{x \in (c_{j-1}, c_j)}$ and $f^+(x) := \sum_{j=1}^t y_j^+ \mathbb{1}_{x \in (c_{j-1}, c_j)}$ the best known lower and upper bounds on the function f inside the boxes B_j. Then, $\left(\sum_{i=1}^t \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon$ is equivalent to

$$\left(\sum_{j=1}^t (y_j^+ - y_j^-)^p \mu\big((c_{j-1}, c_j)\big)\right)^{1/p} \leqslant \varepsilon \,, \qquad \text{that is,} \qquad \|f^+ - f^-\|_p \leqslant \varepsilon \,. \tag{5.2}$$

The last condition $\|f^+ - f^-\|_p \leqslant \varepsilon$ implies that f^- and f^+ are good lower and upper bounds on the function f outside of the points c_j. Note an interesting connection with the definition of bracketing entropy in empirical processes theory (see, e.g., (van der Vaart, 1998, Chapter 19) and references therein).

The following lemma shows that $\mathcal{N}_p(f,\varepsilon)$ is always well defined and at most of the order of $1/\varepsilon$. The proof is postponed to Appendix 5.A, where we collect other useful properties about $\mathcal{N}_p(f,\varepsilon)$.

Lemma 5.1. *For all non-decreasing functions $f : [0,1] \to [0,1]$ and $\varepsilon > 0$, the quantity $\mathcal{N}_p(f,\varepsilon)$ is well defined and upper bounded by*

$$\mathcal{N}_p(f,\varepsilon) \leqslant \lceil 1/\varepsilon \rceil \,. \tag{5.3}$$

Though the rate of $1/\varepsilon$ is tight in the limit $\varepsilon \to 0$ for many functions such as $f : x \mapsto x$ and probability measures such as $\mu = \text{Leb}$, the asymptotic behavior of $\mathcal{N}_p(f,\varepsilon)$ when $\varepsilon \to 0$ does not necessarily reflect the shape of f for a given $\varepsilon \in (0,1]$. Indeed all functions f which are ε_0-close to some piecewise-constant function have a very small box-covering number $\mathcal{N}_p(f,\varepsilon_0)$, even if $\mathcal{N}_p(f,\varepsilon)$ is large in the limit $\varepsilon \to 0$. Importantly, as we show in the next sections, the estimation problem addressed in this paper is very easy for such functions f and scales ε_0.

5.2 Lower bound

In this section we prove a lower bound on the number of evaluations of f that any deterministic algorithm must request in order to guarantee an ε-approximation of f in $L^p(\mu)$-norm after stopping, when only given the prior knowledge that $f : [0,1] \to [0,1]$ is non-decreasing. The next theorem states that in such a case, at least $\mathcal{N}_p(f,2\varepsilon) - 1$ evaluations of f are necessary.

In the sequel we write $\tau(f)$ to make it explicit that the stopping time τ of the algorithm depends on the underlying function f (through the sequentially observed values $f(x_1), f(x_2), \ldots$).

Theorem 5.2. *Let $\varepsilon > 0$ and $(\tau(f), (x_t)_{1 \leqslant t \leqslant \tau(f)}, \widehat{f}_{\tau(f)})$ be the output of any deterministic algorithm such that, for all non-decreasing functions $f : [0,1] \to [0,1]$,*

$$\tau(f) < +\infty \qquad \text{and} \qquad \|\widehat{f}_{\tau(f)} - f\|_p \leqslant \varepsilon \,.$$

Then, for all non-decreasing functions $f : [0,1] \to [0,1]$,

$$\tau(f) \geqslant \mathcal{N}_p(f,2\varepsilon) - 1 \,.$$

In words, any algorithm that is guaranteed to output an ε-approximation after finitely-many evaluations whatever the non-decreasing function f *must* evaluate each f at least $\mathcal{N}_p(f,2\varepsilon) - 1$ times before stopping. In Section 5.3 we show a matching upper bound up to a multiplicative factor of the order of $p\log(1/\varepsilon)$. This indicates that the box-covering number $\mathcal{N}_p(f,2\varepsilon)$ introduced in Section 5.1.2 is a key quantity to describe the inherent difficulty of the estimation problem.

Note that the lower bound holds for every f simultaneously. It thus has a similar flavor to distribution-dependent lower bounds that have been proved for the stochastic multi-armed bandit problem in online learning theory (see, e.g., Chapter 16 by Lattimore and Szepesvári Lattimore and Szepesvári (2020)). Recently an f-dependent lower bound (also based on a notion of cover) was proved by Bachoc et al. (2021) for certified zeroth-order Lipschitz optimization, where algorithms are required to output error certificates (i.e., observable upper bounds on the optimization error).

Proof. Assume for a moment that there exists a non-decreasing function $g : [0,1] \to [0,1]$ such that $\tau(g) < \mathcal{N}_p(g, 2\varepsilon) - 1$. When run on g, the algorithm only uses the $\tau(g)$ query points $x_1, \dots, x_{\tau(g)}$ before stopping. Let $0 < \widetilde{x}_1 < \ldots < \widetilde{x}_n < 1$ denote the ordered values after removing redundancies and the values 0 and 1 (if applicable), with $0 \leqslant n \leqslant \tau(g)$. Consider the adjacent boxes $B_i = [\widetilde{x}_i, \widetilde{x}_{i+1}] \times [g(\widetilde{x}_i), g(\widetilde{x}_{i+1})]$ for $i \in \{0, \dots, n\}$ where we set $\widetilde{x}_0 = 0$ and $\widetilde{x}_{n+1} = 1$. The sequence $B_0, \dots, B_n$ is a box-cover of g (by monotonicity). We construct two functions g_- and g_+ that surround g:

$$g_- : x \mapsto \begin{cases} g(\widetilde{x}_i) & \text{if } \widetilde{x}_i \leqslant x < \widetilde{x}_{i+1} \\ g(1) & \text{if } x = 1 \end{cases} \text{ and } g_+ : x \mapsto \begin{cases} g(\widetilde{x}_{i+1}) & \text{if } \widetilde{x}_i < x \leqslant \widetilde{x}_{i+1} \\ g(0) & \text{if } x = 0\,. \end{cases}$$

Since $g_-(\widetilde{x}_i) = g_+(\widetilde{x}_i) = g(\widetilde{x}_i)$ for all $i \in \{0, \dots, n+1\}$, the algorithm (which is deterministic) would behave the same when run with g_- or g_+ as when run with g, and would construct the same approximation function $\widehat{g}_{\tau(g)}$ after the same number $\tau(g_-) = \tau(g_+) = \tau(g)$ of evaluations. However,

$$\|g_+ - g_-\|_p^p = \sum_{i=0}^{n} (g(\widetilde{x}_{i+1}) - g(\widetilde{x}_i))^p \mu((\widetilde{x}_i, \widetilde{x}_{i+1})) = \sum_{i=0}^{n} \mathcal{A}_p^p(B_i) > (2\varepsilon)^p\,,$$

where the last inequality follows from $n + 1 \leqslant \tau(g) + 1 < \mathcal{N}_p(g, 2\varepsilon)$ and the definition of $\mathcal{N}_p(g, 2\varepsilon)$. The triangle inequality then yields

$$\|\widehat{g}_{\tau(g)} - g_-\|_p + \|\widehat{g}_{\tau(g)} - g_+\|_p \geqslant \|g_+ - g_-\|_p > 2\varepsilon\,,$$

which shows that one of $\|\widehat{g}_{\tau(g)} - g_-\|_p$ or $\|\widehat{g}_{\tau(g)} - g_+\|_p$ is larger than ε. Since (as proved above) $\widehat{g}_{\tau(g)}$ is the approximation function output by the algorithm both with g_- and g_+, which are both non-decreasing, the last conclusion is in contradiction with the assumption that $\|\widehat{f}_{\tau(f)} - f\|_p \leqslant \varepsilon$ for all non-decreasing functions $f : [0,1] \to [0,1]$. This concludes the proof. □

5.3 Upper bound

In this section we introduce the GreedyBox algorithm and derive an f-dependent sample complexity bound (Theorem 5.3) that matches the lower bound of Theorem 5.2 up to a logarithmic factor, for every non-decreasing function f. In Section 5.4 we will study consequences and derive improved bounds for integral estimation (in expectation) and worst-case approximation of piecewise-C^2 functions.

5.3.1 Algorithm and main result

We consider Algorithm 13 below, which draws heavily on an algorithm proposed by Novak (1992, Section 3.2) for numerical integration, and which we call GreedyBox thereafter. It is remarkably simple: at every round, it selects the largest box in the current box-cover of f and replaces it with two smaller boxes by evaluating f at the middle or, more generally, at a conditional median for a general probability measure μ. At any $t \geqslant 1$, we approximate f with the trapezoidal estimator $\widehat{f}_t$ defined as the piecewise-affine function that joins the points $(b_k^t, f(b_k^t))_{0 \leqslant k \leqslant t}$ visited up to time t. Note that this estimator uses $t+1$ evaluations of f. We stop Algorithm 13 at time τ_ε, which is the first $t \geqslant 1$ when the certificate $\xi_t = \sum_{k=1}^t (a_k^t)^p$ falls below ε^p. (This is because ξ_t is a valid upper bound on $\|\widehat{f}_t - f\|_p^p$, by Lemma 5.4 below.)

Algorithm 13 GreedyBox (inspired from Novak 1992, Section 3.2).

Inputs: $\varepsilon \in (0,1], p \geqslant 1$, probability measure μ on $[0,1]$.

Initialization: Set $t = 1$, $x_0 = b_0^1 = 0$, $x_1 = b_1^1 = 1$, evaluate $f(0)$ and $f(1)$, and set $\xi_1 = (a_1^1)^p = (f(1) - f(0))^p \mu((0,1))$.

1: **while** $\xi_t > \varepsilon^p$ **do**

2: Select the box with the largest generalized area: find $k_*^t \in \arg\max_{k \in \{1,\dots,t\}} a_k^t$. over $k \in \{1,\dots,t\}$.

3: Let x_{t+1} be a median of the conditional distribution $\mu(\cdot | (b_{k_*^t - 1}^t, b_{k_*^t}^t))$, and evaluate f at x_{t+1}.

4: Sort the points $x_0, x_1, \dots, x_{t+1}$ in increasing order:
$$b_0^{t+1} = 0 \leqslant b_1^{t+1} \leqslant \dots \leqslant b_t^{t+1} = 1.$$

5: Define the generalized areas for all $k \in \{1,\dots,t+1\}$
$$a_k^{t+1} := \mu((b_{k-1}^{t+1}, b_k^{t+1}))^{1/p} (f(b_k^{t+1}) - f(b_{k-1}^{t+1}))$$

6: Update the certificate
$$\xi_{t+1} = \sum_{k=1}^{t+1} (a_k^{t+1})^p . \tag{5.4}$$

7: Let $t \to t+1$.

8: **end while**

9: Set $\tau_\varepsilon = t$ and approximate f with the piecewise-affine function $\widehat{f}_{\tau_\varepsilon}$ defined by:
$$\forall x \in [0,1] \qquad \widehat{f}_{\tau_\varepsilon}(x) = \frac{f(b_k^{\tau_\varepsilon}) - f(b_{k-1}^{\tau_\varepsilon})}{b_k^{\tau_\varepsilon} - b_{k-1}^{\tau_\varepsilon}} (x - b_{k-1}^{\tau_\varepsilon}) + f(b_{k-1}^{\tau_\varepsilon}),$$
for $k \in \{1,\dots,\tau_\varepsilon\}$ such that $b_{k-1}^{\tau_\varepsilon} \leqslant x \leqslant b_k^{\tau_\varepsilon}$.

10: **Output:** $(\tau_\varepsilon, (x_t)_{1 \leqslant t \leqslant \tau_\varepsilon}, \widehat{f}_{\tau_\varepsilon})$.

Algorithmic complexity. We assume that a median of the conditional distribution $\mu(\cdot | (b_{k_*^t - 1}^t, b_{k_*^t}^t))$ can be computed exactly at every round t. When μ is the Lebesgue measure, it can indeed be computed in closed form: it is the midpoint $(b_{k_*^t - 1}^t + b_{k_*^t}^t)/2$.

For the sake of simplicity, in Algorithm 13 we perform a sort (Step 4) and an argmax operation (Step 2) at each round t, to get the points in increasing order and to choose the box with the largest generalized area. However, one can get rid of the sort operation at each round and do it only once at the end, because GreedyBox does not need the

order of the boxes before the last iteration, where it uses sorted points to build $\widehat{f}_{\tau_\varepsilon}$. Furthermore, for the argmax operation, naive methods yield a computational complexity of $\mathcal{O}(t)$ at each time t, resulting in a quadratic complexity for GreedyBox, far worse than the linear complexity of traditional algorithms such as the trapezoidal rule. To speed up GreedyBox, we use a classical algorithmic trick: a max-heap, which is a binary tree that takes logarithmic time to both remove the maximum value and add an element. This provides GreedyBox with a computational complexity of $\mathcal{O}(t\log(t))$ after t rounds, which is closer to the complexity of the trapezoidal rule.

Upper bound on the sample complexity. Let $\varepsilon \in (0,1]$ be some target accuracy level. The next theorem provides a bound on the sample complexity τ_ε of GreedyBox. The proof appears in Section 5.3.2.

Theorem 5.3. *Let $f : [0,1] \to [0,1]$ be non-decreasing,*[5] *$p \geqslant 1$, and $\varepsilon \in (0,1]$. Then,* GreedyBox *defined above (Algorithm 13) satisfies*

$$\|\widehat{f}_{\tau_\varepsilon} - f\|_p := \left(\int_0^1 (\widehat{f}_{\tau_\varepsilon}(x) - f(x))^p \,\mathrm{d}\mu(x)\right)^{1/p} \leqslant \varepsilon\,,$$

at the stopping time τ_ε. Furthermore, its sample complexity is bounded as follows:

$$\tau_\varepsilon \leqslant 32p^2\big(\log_2(2/\varepsilon^2)+2\big)^2\mathcal{N}_p(f,\varepsilon)\,.$$

We make three comments before proving the theorem.

A new f-dependent bound. Since $\mathcal{N}_p(f,\varepsilon) \leqslant \lceil 1/\varepsilon \rceil$ for all non-decreasing functions f (by Lemma 5.1), the above sample complexity bound $\tau_\varepsilon = \mathcal{O}\left(\mathcal{N}_p(f,\varepsilon)\log^2(1/\varepsilon)\right)$ implies the well-known upper bound of $\mathcal{O}(1/\varepsilon)$ up to logarithmic factors in the worst case. Importantly, though the rate of $1/\varepsilon$ is worst-case optimal (see, e.g., (Kiefer, 1957, Section 5.A)), Theorem 5.3 yields a much better bound for functions f that are easier to approximate, such as functions close to piecewise-constant functions, because $\mathcal{N}_p(f,\varepsilon)$ is small in that case. Since the GreedyBox algorithm uses no prior knowledge on f (beyond monotonicity) to stop at τ_ε, it is adaptive to the unknown complexity $\mathcal{N}_p(f,\varepsilon)$.

A nearly optimal bound. Note that the lower bound of Theorem 5.2 is in terms of $\mathcal{N}_p(f,2\varepsilon)$, while the upper bound of Theorem 5.3 is proportional to $\mathcal{N}_p(f,\varepsilon)$. By a simple argument (dividing boxes p times to reduce their generalized widths by a factor of 2^p, similarly to the proof of Lemma 5.6), we can prove that $\mathcal{N}_p(f,\varepsilon) \leqslant 2^p\mathcal{N}_p(f,2\varepsilon)$. Therefore, the lower and upper bounds of Theorems 5.2 and 5.3 match up to a logarithmic factor. For each non-decreasing function $f : [0,1] \to [0,1]$, GreedyBox is thus nearly optimal among all algorithms with guaranteed $L^p(\mu)$ error after stopping.

A possible minor improvement. When μ is the Lebesgue measure, the bound on τ_ε could be slightly improved (in the constants) by replacing the certificate in Equation (12) with $\xi_{t+1} = \frac{1}{1+p}\sum_{k=1}^{t+1}(a_k^t)^p$ (see Lemma 5.17 in Appendix 5.B.1). While a similar minor improvement is likely to hold for general μ with a slightly different interpolation $\widehat{f}_t$ (non-necessarily piecewise-affine), we decided to focus on piecewise-affine interpolations for the sake of presentation.

[5] Recall that the input and output sets of f can be rescaled to any non-empty compact intervals $\mathcal{X}$ and $\mathcal{Y}$ of $\mathbb{R}$, changing the results only by a multiplicative constant.

5.3.2 Proof of Theorem 5.3

Before proving Theorem 5.3, we first state three lemmas, whose proofs are all postponed to Appendix 5.B.

The first one below shows that, at any round t before stopping, the error of GreedyBox is at most the sum of the generalized areas to the power p of the current-box cover of f. We recall that $a_k^t := (\mu(b_{k-1}^t, b_k^t))^{1/p}(f(b_k^t) - f(b_{k-1}^t))$ denotes the generalized area of the k-th box at round t, and we define the trapezoidal estimato $\widehat{f}_t$ to be the piecewise-affine function that joins the points $(b_k^t, f(b_k^t))_{0 \leqslant k \leqslant t}$ visited up to time t.

Lemma 5.4. *Let $f : [0,1] \to [0,1]$ be non-decreasing, $p \geqslant 1$ and $\varepsilon \in (0,1]$. For any $t \in \{1, \dots, \tau_\varepsilon\}$,*

$$\|\widehat{f}_t - f\|_p^p := \int_0^1 \left|\widehat{f}_t(x) - f(x)\right|^p \mathrm{d}\mu(x) \leqslant \sum_{k=1}^{t} (a_k^t)^p =: \xi_t \,.$$

The next two lemmas are used to control τ_ε. We first show (by a dichotomy argument) that the algorithm can quickly make all boxes equally small. Recall from Section 5.1.2 that $\mathcal{N}_p'(f,\varepsilon)$ denotes the minimum cardinality of a box-cover of f for which each box has a generalized area below ε.

Lemma 5.5. *Let $f : [0,1] \to [0,1]$ be non-decreasing, $p \geqslant 1$ and $\varepsilon \in (0,1]$. Define $\tau_\varepsilon' := 2(1 + \lceil p \log_2(1/\varepsilon) \rceil)\mathcal{N}_p'(f,\varepsilon)$, and assume that* GreedyBox *is such that $\tau_\varepsilon > \tau_\varepsilon'$. Then, at time τ_ε', all the boxes maintained by* GreedyBox *have a generalized area bounded from above by ε, i.e., $a_k^{\tau_\varepsilon'} \leqslant \varepsilon$ for all $k \in \{1, \dots, \tau_\varepsilon'\}$.*

The next lemma shows that the certificate $\xi_t = \sum_{k=1}^t (a_k^t)^p$ at round t decreases at least linearly in t.

Lemma 5.6. *Let $f : [0,1] \to [0,1]$ be non-decreasing, $p \geqslant 1$ and $\varepsilon \in (0,1]$. For any $t \in \{1, \dots, \lfloor \tau_\varepsilon/2 \rfloor\}$, we have $\xi_{2t} \leqslant \xi_t/2$. Therefore, for all $t \leqslant s$ in $\{1, \dots, \tau_\varepsilon\}$,*

$$\xi_s \leqslant \frac{\xi_t}{2^{\lfloor \log_2(s/t) \rfloor}} \leqslant \left(\frac{2t}{s}\right) \xi_t \,. \tag{5.5}$$

Proof of Theorem 5.3. We are now ready to prove the theorem. The first inequality follows immediately from Lemma 5.4 and from the fact that $\xi_{\tau_\varepsilon} \leqslant \varepsilon^p$ by definition of τ_ε.

We now show by contradiction that $\tau_\varepsilon \leqslant 32(1 + \lceil p \log_2(2/\varepsilon^2) \rceil)^2 \mathcal{N}_p(f,\varepsilon)$. Assume thus for a moment that

$$\tau_\varepsilon > 32(1 + \lceil p \log_2(2/\varepsilon^2) \rceil)^2 \mathcal{N}_p(f,\varepsilon) \,. \tag{5.6}$$

This assumption will be used implicitly when calling Lemmas 5.5 and 5.6 below, since it will imply that $\tau_{\varepsilon'}' \leqslant \tau_\varepsilon'' < \tau_\varepsilon \leqslant \tau_{\varepsilon'}$ (so that the algorithm has not stopped before any round considered below). We will see in the end that it raises a contradiction.

Let $n_\varepsilon := \mathcal{N}_p(f,\varepsilon)$. By Lemma 5.5 applied with $\varepsilon' = \varepsilon / n_\varepsilon^{1/p}$, at time $\tau_{\varepsilon'}' := 2(1 + \lceil p \log_2(1/\varepsilon') \rceil)\mathcal{N}_p'(f,\varepsilon')$, the $\tau_{\varepsilon'}'$ boxes maintained by GreedyBox all have generalized areas at most of ε' each, so that the certificate $\xi_{\tau_{\varepsilon'}'}$ satisfies

$$\begin{aligned} \xi_{\tau_{\varepsilon'}'} &\leqslant \tau_{\varepsilon'}' \cdot (\varepsilon')^p \leqslant 2(1 + \lceil p \log_2(1/\varepsilon') \rceil)\mathcal{N}_p'(f,\varepsilon') \cdot \frac{\varepsilon^p}{n_\varepsilon} \\ &\leqslant 4(1 + \lceil p \log_2(2/\varepsilon^2) \rceil)\, \varepsilon^p \,, \end{aligned} \tag{5.7}$$

where we used the fact that $\mathcal{N}'_p(f,\varepsilon') \leqslant 2\mathcal{N}_p(f,\varepsilon)$ (by Lemma 5.16 in Appendix 5.A.3) and that $1/\varepsilon' = \mathcal{N}_p(f,\varepsilon)^{1/p}/\varepsilon \leqslant 2^{1/p}/\varepsilon^{(p+1)/p} \leqslant 2/\varepsilon^2$ (since $\mathcal{N}_p(f,\varepsilon) \leqslant \lceil 1/\varepsilon \rceil \leqslant 2/\varepsilon$). Now, we apply Lemma 5.6 with $t = \tau'_{\varepsilon'}$ and $s = \tau''_\varepsilon := 8(1+\lceil p\log_2(2/\varepsilon^2)\rceil)\tau'_{\varepsilon'}$. It yields:

$$\xi_{\tau''_\varepsilon} \leqslant \left(\frac{2\tau'_{\varepsilon'}}{8(1+\lceil p\log_2(2/\varepsilon^2)\rceil)\tau'_{\varepsilon'}}\right)\xi_{\tau'_{\varepsilon'}} \overset{\text{by (5.7)}}{\leqslant} \varepsilon^p\,.$$

This raises a contradiction with (5.6), since (using again $\mathcal{N}'_p(f,\varepsilon') \leqslant 2\mathcal{N}_p(f,\varepsilon)$)

$$\tau''_\varepsilon = 8(1+\lceil p\log_2(2/\varepsilon^2)\rceil)\tau'_{\varepsilon'} \leqslant 32(1+\lceil p\log_2(2/\varepsilon^2)\rceil)^2\mathcal{N}_p(f,\varepsilon)$$

and τ_ε is by definition the first time t such that $\xi_t \leqslant \varepsilon^p$. Therefore, (5.6) must be false, so that $\tau_\varepsilon \leqslant 32(1+\lceil p\log_2(2/\varepsilon^2)\rceil)^2\mathcal{N}_p(f,\varepsilon)$. Elementary calculations conclude the proof of Theorem 5.3. □

5.4 Improvement for special cases

In this section, we derive consequences (with rates faster than the worst-case ε^{-1}) in two specific cases: integral estimation and piecewise-smooth functions.

In the sequel, we adopt a slightly different yet equivalent viewpoint than in Section 5.3. Though Algorithms 14 and 15, defined in this section, formally stop at round τ_ε, in the proofs, we extend their definitions to all rounds $t \geqslant 1$, by replacing the while condition with $\xi_t > 0$, defining τ_0 as the first round t (if any) where the certificate ξ_t reaches 0, and setting $\widehat{f}_s := \widehat{f}_{\tau_0}$ for all subsequent rounds $s \geqslant \tau_0 + 1$. Note that their approximation error equals zero for all $s \geqslant \tau_0$.

5.4.1 Side problem: integral estimation

Throughout this subsection, we focus on the case of integral estimation rather than approximation in $L^p(\mu)$-norm. The goal is to approximate the integral $I(f) = \int_0^1 f(x)\,\mathrm{d}\mu(x)$ of a non-decreasing function f on $[0,1]$. This problem is simpler than the $L^1(\mu)$-approximation problem studied previously, and thus GreedyBox can be easily extended to integral estimation while maintaining the same bound.

A deterministic algorithm for the integral estimation problem is defined as a procedure that, given $\varepsilon > 0$, produces a tuple $(\tau_\varepsilon, (x_t)_{1\leqslant\tau_\varepsilon}, \widehat{I}_{\tau_\varepsilon}(f)) \in \mathbb{N}_+ \times \mathcal{X}^{\tau_\varepsilon} \times \mathbb{R}$, where $(x_t)_{t\geqslant 1}$ and τ_ε are defined sequentially, similar to the approximation in $L^p(\mu)$-norm. That is: for all $t \geqslant 2$, x_t is a measurable function of the history h_{t-1} and τ_ε is a stopping time after which the process of the algorithm ends. The algorithm finally outputs an approximation $\widehat{I}_{\tau_\varepsilon}(f)$ of the integral. We now study the convergence speed of GreedyBox in this setting (we replace the definition of $\widehat{f}_{\tau_\varepsilon}$ in the last line of GreedyBox by the computation of $\widehat{I}_{\tau_\varepsilon}(f) = \int_0^1 \widehat{f}_{\tau_\varepsilon}(x)\,\mathrm{d}\mu(x)$) and check whether it achieves optimal convergence speed.

Nearly optimal performance for integral estimation

The following upper bound on the sample complexity of GreedyBox is a direct consequence of Theorem 5.3 with $p = 1$.

Corollary 5.7. *Let $f:[0,1]\to[0,1]$ be non-decreasing, and $\varepsilon\in(0,1]$. Then,* GreedyBox *with $p=1$ satisfies, at the stopping time τ_ε,*

$$\left|\int_0^1 \widehat{f}_{\tau_\varepsilon}(x)\,\mathrm{d}\mu(x)-\int_0^1 f(x)\,\mathrm{d}\mu(x)\right|\leqslant\varepsilon\,.$$

Besides, its sample complexity is bounded from above as follows:

$$\tau_\varepsilon\leqslant 32p^2\big(\log_2(2/\varepsilon^2)+2\big)^2\mathcal{N}_p(f,\varepsilon)\,.$$

The question is now to check if the lower bound for this weaker problem is still the same one. The following theorem asserts that this is the case.

Theorem 5.8 (Matching lower bound)**.** *Let $\varepsilon>0$ and $\mathcal{A}$ be any deterministic algorithm such that, for all non-decreasing functions $f:[0,1]\to[0,1]$:*

$$\tau_\varepsilon(f)<+\infty \text{ and } \left|\widehat{I}_{\tau_\varepsilon(f)}(f)-\int_0^1 f(x)\,\mathrm{d}\mu(x)\right|\leqslant\varepsilon\,.$$

Then, for all non-decreasing functions $f:[0,1]\to[0,1]$

$$\tau_\varepsilon(f)\geqslant\mathcal{N}_p(f,2\varepsilon)-1\,.$$

The proof closely follows that of Theorem 5.2 in the case of $p=1$ and is left to the reader. Note that it is inspired from that of the well-known minimax lower bound of $1/(2n+2)$. This lower bound actually implies the lower bound of Theorem 5.2 for $p=1$.

Improvement in expectation with randomization

We now provide a stochastic version of our algorithm; see Algorithm 14 below, which we call StochasticGreedyBox. With this randomized variant we prove better guarantees (in expectation only)[6] for the integral estimation problem than with the deterministic version. Note that the improvement is not true for estimating f in $L^p(\mu)$-norm. The idea to use randomization to improve the rates is due to Novak (1992, Section 2.2); we adapt this idea to a fully sequential algorithm, whose bound is now adaptive to the complexity of f. It's worth mentioning that, for the sake of simplicity, the random points B_k in Algorithm 14 are currently sampled at the conclusion of the algorithm. However, an alternative approach could involve sequential sampling when the intervals (b_{k-1}^t,b_k^t) are created, in order to get a sequential estimator $\widehat{I}_t(f)$ for all $t\geqslant 1$.

The following lemma shows that the error of StochasticGreedyBox (Algorithm 14) at stopping time is indeed at most ε.

Lemma 5.9. *Let $\varepsilon>0$ and let $f:[0,1]\to[0,1]$ be non-decreasing. Then, the output of Algorithm 14 satisfies*

$$\mathbb{E}\Big[\big|\widehat{I}_{\tau_\varepsilon}(f)-I(f)\big|\Big]\leqslant\xi_{\tau_\varepsilon}:=\frac{1}{2}\sqrt{\sum_{k=1}^{\tau_\varepsilon}(a_k^{\tau_\varepsilon})^2}\leqslant\varepsilon\,.$$

[6] We could easily derive high probability bound using Hoeffding's lemma.

Algorithm 14 StochasticGreedyBox.

Inputs: $\varepsilon > 0$, probability measure μ on $[0,1]$.

Initialization: Set $t = 1$, $x_0 = b_0^1 = 0$ and $x_1 = b_1^1 = 1$, evaluate $f(0)$ and $f(1)$ and set $\xi_1 = a_1^1/2 = \mu((0,1))(f(1) - f(0))/2$.

1: **while** $\xi_t > \varepsilon$ **do**

2: Select the box with the largest area: find $k_*^t \in \arg\max_{1 \leqslant k \leqslant t} a_k^t$.

3: Let x_{t+1} be a median of the conditional distribution $\mu(\cdot|(b_{k_*^t-1}^t, b_{k_*^t}^t))$ and evaluate f at x_{t+1}.

4: Sort the points $x_0, x_1, \ldots, x_{t+1}$ in increasing order:
$$b_0^{t+1} = 0 < b_1^{t+1} < \cdots < b_{t+1}^{t+1} = 1\,.$$

5: Define the boxes areas for $k \in \{1, \ldots, t+1\}$ by
$$a_k^{t+1} := \mu((b_{k-1}^{t+1}, b_k^{t+1}))(f(b_k^{t+1}) - f(b_{k-1}^{t+1}))\,.$$

6: Update the certificate
$$\xi_{t+1} = \frac{1}{2}\sqrt{\sum_{k=1}^{t+1}(a_k^{t+1})^2}\,.$$

7: Let $t \leftarrow t+1$.

8: **end while**

9: Set $\tau_\varepsilon = t$ and let $S_{\tau_\varepsilon} = \left\{k \in \{1, \ldots, \tau_\varepsilon\} \text{ s.t. } \mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon})) > 0\right\}$.

10: **for** $k \in S_{\tau_\varepsilon}$ **do**

11: Sample B_k according to μ conditionally to the interval $(b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon})$ and evaluate f at B_k.

12: **end for**

13: Approximate $I(f) = \int_0^1 f(x) d\mu(x)$ with the estimator:
$$\widehat{I}_{\tau_\varepsilon}(f) := \sum_{k \in S_{\tau_\varepsilon}} \mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon}))f(B_k) + \sum_{k=0}^{\tau_\varepsilon} \mu(\{b_k^{\tau_\varepsilon}\})f(b_k^{\tau_\varepsilon})\,.$$

14: **Output:** $(\tau_\varepsilon, (x_t)_{1 \leqslant t \leqslant \tau_\varepsilon}, \widehat{I}_{\tau_\varepsilon}(f))$.

Proof. First, we remark that the stopping time τ_ε and the ordered sequence $b_0^{\tau_\varepsilon}, \ldots, b_{\tau_\varepsilon}^{\tau_\varepsilon}$ are deterministic and do not depend on the randomness of the algorithm. These deterministic points being set, the random variables B_k for $k \in S_{\tau_\varepsilon}$ are distributed according to μ conditionally to each of the deterministic intervals defined by the $b_k^{\tau_\varepsilon}$'s and satisfy for all $k \in S_{\tau_\varepsilon}$

$$B_k \sim \mu(\cdot|(b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon})) \quad \text{and} \quad \mathbb{E}[f(B_k)] = \frac{1}{\mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon}))} \int_{b_{k-1}^{\tau_\varepsilon}}^{b_k^{\tau_\varepsilon}} f(x)\,\mathrm{d}\mu(x)\,.$$

Therefore,

$$\begin{aligned}\mathbb{E}[\widehat{I}_{\tau_\varepsilon}(f)] &= \sum_{k\in S_{\tau_\varepsilon}} \mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon}))\mathbb{E}[f(B_k)] + \sum_{k=0}^{\tau_\varepsilon} \mu(\{b_k^{\tau_\varepsilon}\})f(b_k^t) \\ &= \sum_{k=1}^{\tau_\varepsilon} \int_{(b_{k-1}^t, b_k^t)} f(x)\,\mathrm{d}\mu(x) + \sum_{k=0}^{\tau_\varepsilon} \mu(\{b_k^t\})f(b_k^t) \\ &= \int_0^1 f(x)\,\mathrm{d}\mu(x) = I(f)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\Big[\big|\widehat{I}_{\tau_\varepsilon}(f) - I(f)\big|\Big]^2 &\overset{\text{Jensen}}{\leqslant} \mathbb{E}\Big[\big(\widehat{I}_{\tau_\varepsilon}(f) - I(f)\big)^2\Big] = \operatorname{Var}\big(\widehat{I}_{\tau_\varepsilon}(f)\big) \\ &\overset{\text{Independence}}{=} \sum_{k\in S_{\tau_\varepsilon}} \mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon}))^2 \operatorname{Var}\big(f(B_k)\big)\,. \end{aligned} \tag{5.8}$$

But since $B_k \in (b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon})$, by monotonicity of f, $f(B_k)$ takes values into the interval $[f(b_{k-1}^{\tau_\varepsilon}), f(b_k^{\tau_\varepsilon})]$. Thus,

$$\operatorname{Var}\big(f(B_k)\big) \leqslant \frac{1}{4}\big(f(b_k^{\tau_\varepsilon}) - f(b_{k-1}^{\tau_\varepsilon})\big)^2 = \frac{(a_k^{\tau_\varepsilon})^2}{4\mu((b_{k-1}^{\tau_\varepsilon}, b_k^{\tau_\varepsilon}))^2}$$

by definition of $a_k^{\tau_\varepsilon}$ (see step 3 of Algorithm 14). Therefore, substituting into Inequality (5.8) and taking the square root, we get

$$\mathbb{E}\Big[\big|\widehat{I}_{\tau_\varepsilon}(f) - I(f)\big|\Big] \leqslant \frac{1}{2}\sqrt{\sum_{k=1}^{\tau_\varepsilon} (a_k^{\tau_\varepsilon})^2}\,,$$

which is smaller than ε by the stopping criterion. □

Remark that if all boxes at time τ_ε have similar areas $a_k^{\tau_\varepsilon} \approx a$ (which the algorithm aims at getting by splitting only largest boxes), Lemma 5.9 is asking a to be at most of order $\mathcal{O}(\varepsilon/\sqrt{\tau_\varepsilon})$ in Algorithm 14, while Algorithm 13 required $\mathcal{O}(\varepsilon/\tau_\varepsilon)$. Therefore, this leads to an earlier stopping criterion and smaller sample complexity. Typically, after t rounds, the approximation error of Algorithm 14 is better than the one of Algorithm 13 by a factor $\sqrt{t}$. This is however not so simple to formulate in terms of sample complexity. We will thus only formulate the analog of Theorem 5.3 under the following assumption.

Assumption 6. Let $\varepsilon > 0$. There exist $C > 0$ and $0 < \alpha < 1$ such that $\mathcal{N}(f, \varepsilon_1) \leqslant C\varepsilon_1^{-\alpha}$ for all $\varepsilon_1 \geqslant \varepsilon$.

It is worth to notice that Assumption 6 is mild since $\mathcal{N}(f, \varepsilon) \leqslant \varepsilon^{-1}$ for any non-decreasing f and $\varepsilon > 0$. Furthermore, the assumption is non-asymptotic in ε since the requirement is only for $\varepsilon_1 \geqslant \varepsilon$.

Theorem 5.10. *Let $f : [0,1] \to [0,1]$ be non-decreasing which satisfies Assumption 6 for some $C, \alpha > 0$. Let $\varepsilon > 0$, then Algorithm 14 satisfies $\mathbb{E}[|\widehat{I}_{\tau_\varepsilon}(f) - I(f)|] \leqslant \varepsilon$. Besides the number of function evaluations is bounded from above by*

$$\tau_\varepsilon = \mathcal{O}\big(\log(1/\varepsilon)^{3/2}\varepsilon^{-\frac{1}{1/\alpha+1/2}}\big)\,.$$

The proof is postponed to Appendix 5.B.4. The benefit of the stochastic algorithm is thus to replace the rate $\varepsilon^{-\alpha}$ obtained with the deterministic version with $\varepsilon^{-\frac{1}{1/\alpha+1/2}}$. This result generalizes the one obtained by Novak (1992) in the case of $\alpha = 1$ (which corresponds our worst-case scenario) for a similar algorithm. A notable advantage of our algorithm over Novak's is its fully sequential nature, whereas Novak's algorithm requires a final step that necessitates $\mathcal{O}(\tau_\varepsilon)$ evaluations.

5.4.2 An intriguing result for piecewise-regular functions

For the sake of simplicity, in this section, we restrict ourselves to the *Lebesgue measure*.

As seen before, GreedyBox has a worst-case sample-complexity of order ε^{-1} up to logarithmic factors. An interesting question is how the error rate improves with regularity for non-decreasing functions, as well as how to adapt GreedyBox to achieve optimal rates for piecewise-smooth functions. More precisely, unlike the rest of the paper, in this section, we focus on analyzing the *effective* L^p-error rate of algorithms when run on piecewise-smooth functions. We control the number of evaluations of f until $\|\widehat{f}_t - f\|_p$ falls below ε, rather than the number τ_ε of evaluations until the certificate ξ_t falls below ε^p. The first (classical) complexity quantity can be much smaller than τ_ε (since the algorithm is only aware that f is non-decreasing, and lacks any prior regularity knowledge). This does not contradict the lower bound of Theorem 5.2 and reveals the effective performance of algorithms when run on simpler functions. For instance, on C^2 functions, the trapezoidal rule has an effective rate of order $\varepsilon^{-1/2}$ (a classical result recalled in Appendix 5.B.8), but cannot guarantee ε-accuracy before order ε^{-1} evaluations if only given the knowledge that the underlying function is non-decreasing.

Upper bound on GreedyBox effective sample complexity. We aim to explore an intriguing question: can we establish these guarantees for GreedyBox without making any modifications? Moreover, the trapezoidal rule fails to adapt to piecewise-C^2 functions, even for simple ones such as $f(x) = \mathbb{1}_{x \geqslant 1/3}$. On the contrary, GreedyBox adapts very well to the discontinuities: it converges exponentially fast for any piecewise-constant function. With this in head, one could ask if GreedyBox learns the jumps quickly enough to ensure an ε-accurate L^p error within $\widetilde{\mathcal{O}}(\varepsilon^{-1/2})$ sample-complexity for piecewise-C^2 functions.[7] The next theorem shows that the rate can indeed be improved as long as the number of C^1-singularities[8] is at most of order $\mathcal{O}(\varepsilon^{-1})$. It should be noted that the number of C^1-singularities may explode to infinity as ε approaches zero and that the number of C^2-singularities does not affect the upper bound.

Theorem 5.11. *Let $\alpha > 0$ and $\varepsilon \in (0,1]$. Let $f : [0,1] \to [0,1]$ be a non-decreasing and piecewise-C^2 function with a number of C^1-singularities bounded by $\varepsilon^{-\alpha}$ and such that $|f''(x)| \leqslant 1$ whenever it is defined. Then, there exists*

$$t_\varepsilon = \begin{cases} \widetilde{\mathcal{O}}\left(\varepsilon^{-1+\frac{1}{2p+2}}\right) & \text{if } \alpha \leqslant \frac{1}{2} \\ \widetilde{\mathcal{O}}\left(\varepsilon^{-1+\left(\frac{1-\alpha}{1+p}\right)_+}\right) & \text{if } \alpha \geqslant \frac{1}{2} \end{cases}$$

[7] The notation $\widetilde{\mathcal{O}}$ hides logarithmic factors.

[8] We call C^k-singularity of f a point $x \in [0,1]$ such that f is not C^k on any neighborhood of x. A discontinuity is a C^0-singularity.

such that $\|\widehat{f}_t - f\|_p \leqslant \varepsilon$ *for all* $t \geqslant t_\varepsilon$, *where* $\widehat{f}_t$ *is the approximation of* f *returned by* GreedyBox *after* t *rounds.*

Theorem 5.11 shows that, for piecewise-C^2 functions with $\alpha < 1$, GreedyBox achieves ε-accuracy in $o(\varepsilon^{-1})$ function evaluations which improves the worst-case guarantee of Theorem 5.3. In particular, when the number of singularities is finite, then $\alpha \to 0$ when $\varepsilon \to 0$, and the L^1-error is asymptotically of order $\mathcal{O}(\varepsilon^{-3/4})$. Note that this result contrasts with what happens for piecewise-C^2 functions without the non-decreasing assumption considered by Plaskota et al. (2008), who showed that as soon as there is strictly more than one discontinuity, any algorithm has a worst-case L^p-error of order $\Omega(\varepsilon^{-p})$.

Minimax lower bound for approximating non-decreasing piecewise-smooth functions. Interestingly, we now show that this result is optimal (up to logarithmic factors) among deterministic algorithms in the regime $\alpha \geqslant 1/2$ (Proposition 5.12), that is when the number of C^1-singularities is at least of order $\Omega(\varepsilon^{-1/2})$. In the other regime, which corresponds to more regular functions, we also show that our upper bound on GreedyBox cannot be improved (Proposition 5.13).

Proposition 5.12. *Let* $p \geqslant 1$, $\varepsilon \in (0,1)$ *and* $\alpha > 0$. *Then, for any deterministic adaptive algorithm* $\mathcal{A}$ *and for any*

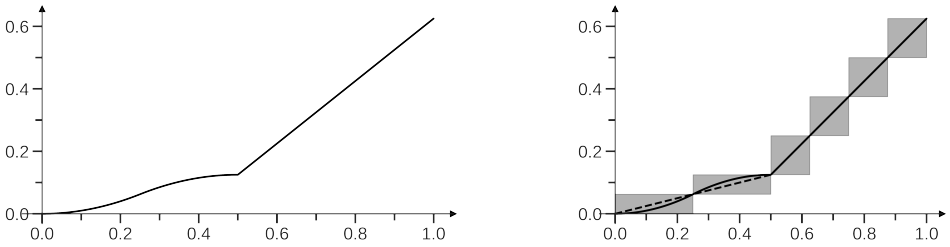
$$t < (2\varepsilon)^{-1+\left(\frac{1-\alpha}{1+p}\right)_+} - 1\,,$$

there exists a non-decreasing piecewise-affine function $f : [0,1] \to [0,1]$ *with at most* $\max\{2, \lceil \varepsilon^{-\alpha} \rceil\}$ *discontinuities, such that* $\|f - \widehat{f}_t\|_p > \varepsilon$.

Remarkably, the above lower bound demonstrates a clear difference between regular functions and piecewise-regular functions, even when the number of pieces is finite. Specifically, when considering the case of $p = 1$, the above lower bound shows that, for piecewise-C^∞ function with a constant number of discontinuities ($\alpha = 0$), surpassing the bound of $\Omega(\varepsilon^{-1/2})$ is not achievable. This demonstrates the influence of singularities, as C^k functions with no singularities can be approximated at a rate $\varepsilon^{-1/k}$. It is also worth pointing out that the above lower bound may be easily extended to non-adaptive algorithms (considering Heaviside step adversarial functions) which would require $\Omega(\varepsilon^{-1})$ function evaluations. This underscores, in a new scenario, the need of adaptive algorithms to approximate functions with singularities Plaskota et al. (2008).

Negative result for GreedyBox. The previous result shows that GreedyBox is (up to logs) optimal for highly non-regular functions ($\alpha \geqslant 1/2$). We now consider the other regime ($\alpha < 1/2$) and prove an almost (up to logs) matching lower bound in the case $p = 1$, $\alpha = 0$. This, shows that the rate $\varepsilon^{-3/4}$ cannot be improved for GreedyBox for such classes of functions.

Proposition 5.13. *Let* $\varepsilon \in (0, 1/12)$. *Then, there exists a piecewise-*C^2 *function* $f_\varepsilon : [0,1] \to [0,1]$ *with* $|f''(x)| \leqslant 1$ *whenever it is defined and one* C^1*-singularity, such that there exists* $t \geqslant 2^{-7}\varepsilon^{-3/4}$ *with* $\|\widehat{f}_t - f_\varepsilon\|_1 > \varepsilon$, *where* $\widehat{f}_t$ *is the* GreedyBox *approximation after* t *rounds.*



(a) Worst-case function f_ε.

(b) Box-cover produced by GreedyBox.

Figure 5.2: Plot of the worst-case function f_ε for $\varepsilon = 0.05$ and box-cover produced by GreedyBox after $t = 6$ iterations.

An example of the worst-case function f_ε built in the proof of Proposition 5.13 for $\varepsilon = 0.005$, that exhibits poor performance for GreedyBox at $t = 6$, is depicted in Figure 5.2. The function is formally defined in the proof and consists of two parts: one that oscillates with $|f''(t)| = 1$ for $x \leqslant 1/2$, and the other that is linear for $x > 1/2$. The function is constructed in such a way that at a certain t ($t = 6$ in Fig. 5.2), GreedyBox selects its points precisely between the oscillations, and focuses too much on the linear part, resulting in a maximum possible L^1 error. It is worth noting that for each value of ε, it is possible to construct an adversarial function f_ε. However, an interesting question arises: can a single function f be devised to work uniformly for all values of ε? We believe that this question is challenging and connected to the 10^{th} open problem raised in (Brass and Petras, 2011, Chapter 10).

GreedyWidthBox: an optimal modification of GreedyBox. GreedyBox can actually be adapted to achieve the optimal rate of $\widetilde{\mathcal{O}}(\varepsilon^{-1+(\frac{1-\alpha}{1+p})_+})$ simultaneously for all $\alpha \geqslant 0$, while maintaining our adaptive guarantee in terms of $\mathcal{N}_p(f, \varepsilon)$. This can be accomplished by employing the GreedyBox approach for half of the iterations and the trapezoidal rule for the remaining half (see Algorithm 15). The proof, left to the reader, follows closely the one of Theorem 5.11, in which the upper bound (5.17) can be simplified by utilizing the fact that, after conducting t function evaluations, the trapezoidal rule ensures that all widths w_i are at most of the order t^{-1}. In particular, for $p = 1$ and $\alpha = 0$, this provides an algorithm that achieves ε-acccuracy in L^1 norm in $\widetilde{\mathcal{O}}(\varepsilon^{-1/2})$ function evaluations for non-decreasing piecewise-C^2 functions, as soon as the number of C^1-singularities remains constant as $\varepsilon \to 0$.

5.5 Numerical experiments

In this section, we study empirically the performance of GreedyBox as compared to the trapezoidal rule. All the experiments are run using $p = 1$ and the Lebesgue measure for μ. Figure 5.3 displays the output given by both GreedyBox and the trapezoidal rule on the function f defined by $f(x) = \frac{1}{2}x^{3/10}$ if $x \leqslant \frac{2}{3}$ and $f(x) = x$ otherwise. One can see that on this example the L^1 error of GreedyBox is twice as small as that of the trapezoidal rule. In general, GreedyBox copes far better with discontinuities than the trapezoidal rule.

Algorithm 15 GreedyWidthBox

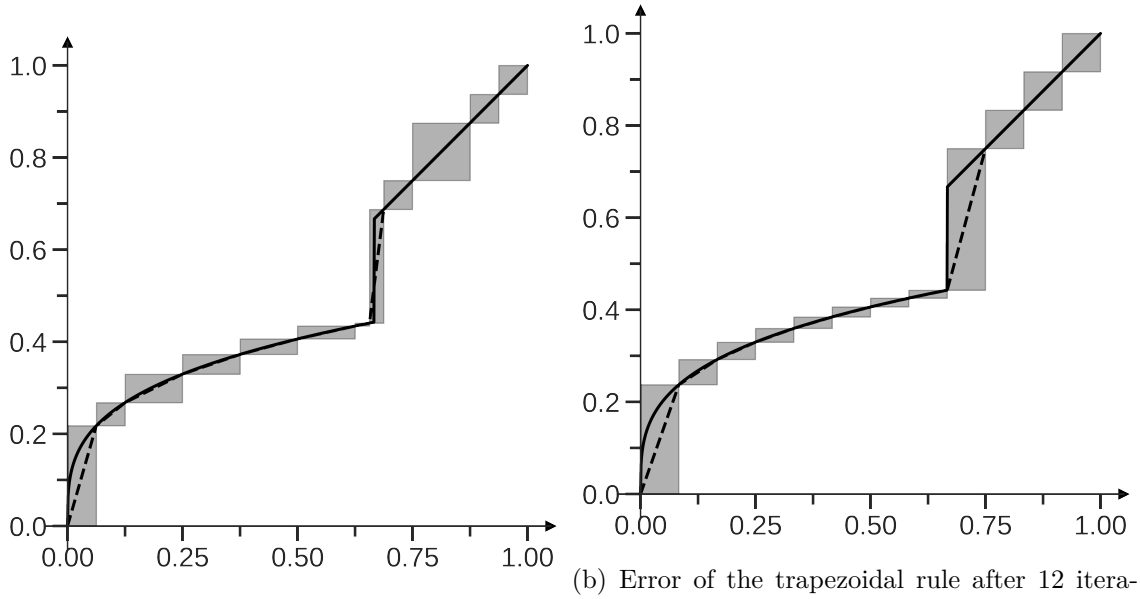
Inputs: $\varepsilon > 0$, $p \geqslant 1$

Initialization: Set $t = 1$, $x_0 = b_0^1 = 0$ and $x_1 = b_1^1 = 1$, evaluate $f(0)$ and $f(1)$, and define $\xi_1 = (a_1^1)^p = (f(1) - f(0))^p$.

1: **while** $\xi_t > \varepsilon^p$ **do**
2: **if** t is even **then**
3: Select the box with the largest width: find k_*^t that maximizes $(b_k^t - b_{k-1}^t)$.
4: **else**
5: Select the box with the largest area: find k_*^t that maximizes a_k^t.
6: **end if**
7: Evaluate f at the midpoint $x_{t+1} := (b_{k_*^t-1}^t + b_{k_*^t}^t)/2$.
8: Sort the points $x_0, x_1, \ldots, x_{t+1}$ in increasing order:
$$b_0^{t+1} = 0 \leqslant b_1^{t+1} \leqslant \ldots \leqslant b_{t+1}^{t+1} = 1;$$
9: Define the generalized area for all $k \in \{1, \ldots, t+1\}$ by
$$a_k^{t+1} = (b_k^{t+1} - b_{k-1}^{t+1})^{1/p} (f(b_k^{t+1}) - f(b_{k-1}^{t+1})).$$
10: Update the certificate
$$\xi_{t+1} = \sum_{k=1}^{t+1} (a_k^{t+1})^p$$
11: Let $t \to t+1$.
12: **end while**
13: et $\tau_\varepsilon = t$ and approximate f with the piecewise-linear function $\widehat{f}_{\tau_\varepsilon}$ defined by:
$$x \in [0,1] \qquad \widehat{f}_{\tau_\varepsilon}(x) = \frac{f(b_k^{\tau_\varepsilon}) - f(b_{k-1}^{\tau_\varepsilon})}{b_k^{\tau_\varepsilon} - b_{k-1}^{\tau_\varepsilon}} (x - b_{k-1}^{\tau_\varepsilon}) + f(b_{k-1}^{\tau_\varepsilon}),$$
for $k \in \{1, \ldots, \tau_\varepsilon\}$ such that $b_{k-1}^{\tau_\varepsilon} \leqslant x \leqslant b_k^{\tau_\varepsilon}$.
14: **Output**: $(\tau_\varepsilon, (x_t)_{1 \leqslant t \leqslant \tau_\varepsilon}, \widehat{f}_{\tau_\varepsilon})$

Remark that the trapezoidal rule is usually an offline algorithm that needs the total number t of iterations from the beginning. Fortunately, it can easily be adapted to an online version built on the same model as GreedyBox. Instead of choosing the box with the largest area on Step 2 of GreedyBox, it picks the box with the largest width. This online version matches exactly the offline trapezoidal rule whenever t is a power of 2 and allows for a better comparison with GreedyBox. It is this online version that we use in the next experiments.

The trapezoidal rule is known to have an L^1 error that decreases linearly with the number t of iterations for any non-decreasing functions. This is the same speed of convergence that we proved for GreedyBox in Theorem 5.3. However, for C^2 functions, the L^1 error of the trapezoidal rule decreases quadratically with the number of iterations, which corresponds to a sample complexity $\varepsilon^{-1/2}$. This is better than the upper bound in $\varepsilon^{-3/4}$ proven in Theorem 5.11 for GreedyBox. Remember however than no lower bound of order $\varepsilon^{-3/4}$ was proved so far for GreedyBox on C^2 functions with no singularities. Also note that the trapezoidal rule achieves a rate of $\varepsilon^{-1/2}$ only for C^2 functions, but can have an



(a) Error of GreedyBox after 12 iterations: 0.005

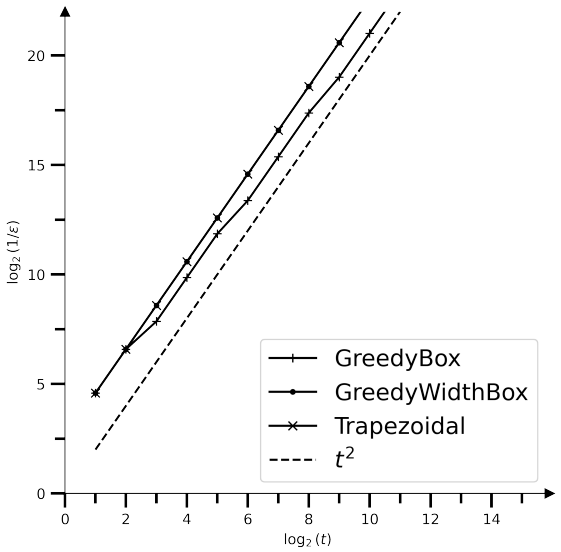
(b) Error of the trapezoidal rule after 12 iterations: 0.015

Figure 5.3: Comparison of GreedyBox and the trapezoidal rule on a piecewise-C^2 function after 12 iterations

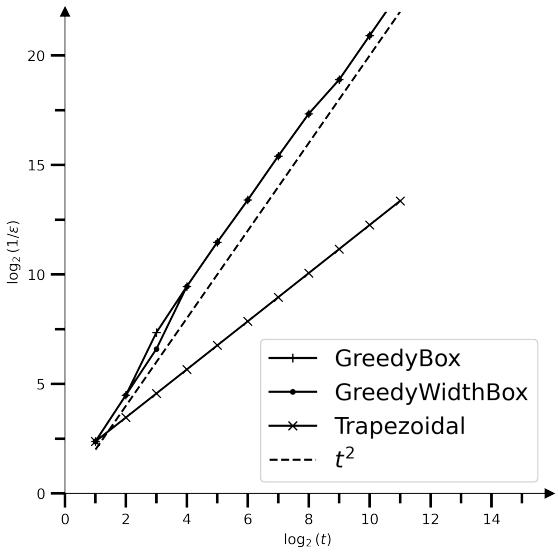
error of order ε^{-1} as soon as the function is discontinuous (see Figure 5.4c).

In order to have best-of-both-worlds theoretical results, we introduced a new algorithm called GreedyWidthBox (Algorithm 15), which achieves the asymptotic rate $\mathcal{O}(\varepsilon^{-1/2})$ for piecewise-C^2 functions, as soon as the number of pieces is finite. In Figure 5.4, we run the three algorithms for a fixed number t of epochs, and observe the L^1 distance between the approximated function $\widehat{f}_t$ returned by the algorithm and the true function. Figures 5.4a-5.4c consider three different functions f with various regularities. In Figure 5.4d, we examine time-dependent piecewise-C^2 functions $g^t : x \mapsto \frac{1}{2} f^t(2x)\mathbb{1}_{x \leqslant 1/2} + \mathbb{1}_{x > 1/2}$. Here, f^t corresponds to the worst-case function as defined in (5.25) in the proof of Proposition 5.13, with an additional introduced discontinuity. For a better understanding of the results, we plot the inverse of this error, and plot everything with logarithmic scale. This means that a straight line with slope 1 represents a linear speed of convergence: an error that decreases inversely proportionally with the number t of epochs. Figure 5.4d precisely confirms the anticipated worst-case rates as determined by the analysis for monotone piecewise-C^2 functions.

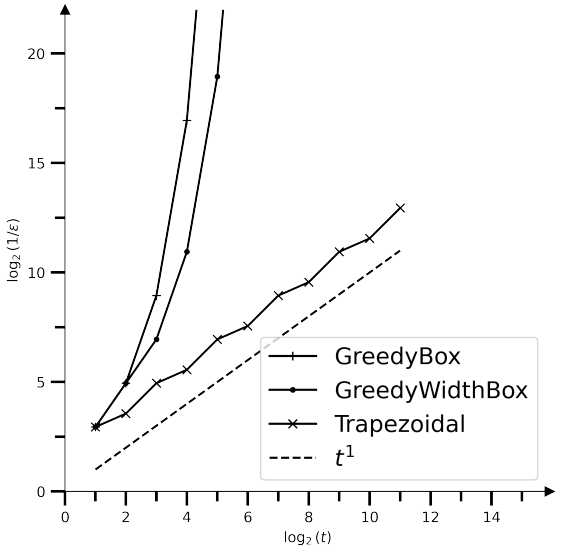
Remember however that given a desired precision ε, GreedyBox does not stop when its L^1 error is smaller than ε, but when its certificate (the best upper bound it can get without knowing *a priori* the function) is smaller than ε. Thus, for computation comparison, what really matters is to plot the convergence of the certificate with regard to some target error ε. The certificate of GreedyBox appears to be smaller than both GreedyWidthBox and the trapezoidal rule, regardless of the smoothness of the function. Yet, for most of the existing functions, the certificate of GreedyBox is of the order of ε^{-1}. However, remark that the same bound apply both for the trapezoidal rule and for GreedyWidthBox. This behavior can be seen on Figure 5.5 for GreedyBox and the trapezoidal rule. In Figure 5.5d,



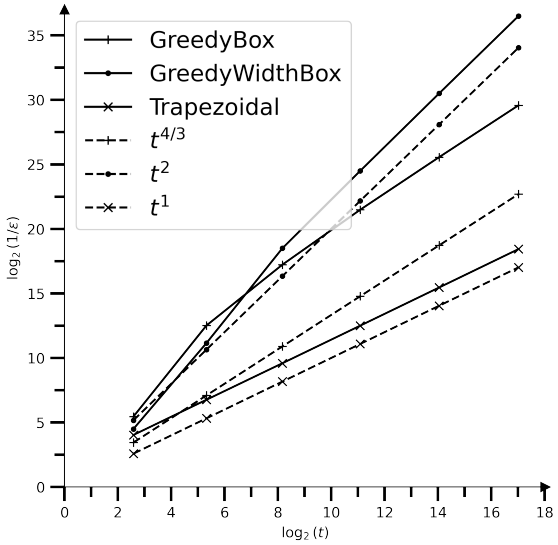
(a) Error rate on $f\colon x \mapsto x^2$



(b) Error rate on $f\colon x \mapsto x^{1/10}$



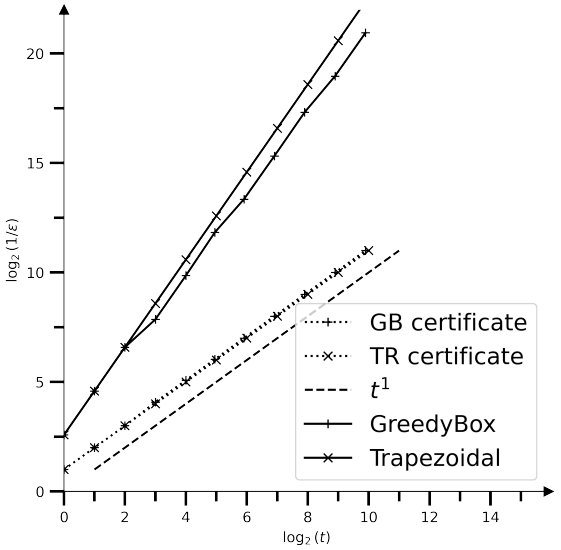
(c) Error rate on $f(x) = \mathbb{1}_{\{x \geqslant 0.3\}}$



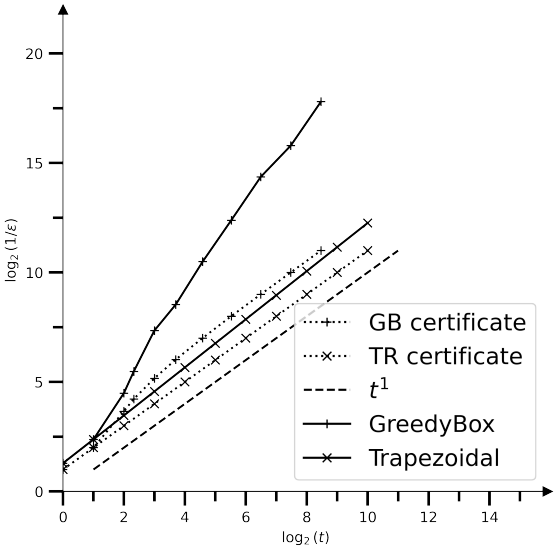
(d) Error rate on g^t

Figure 5.4: Comparison of GreedyBox and GreedyWidthBox with the trapezoidal rule. Logarithmic scale of the inverse of the error w.r.t. the number of evaluations.

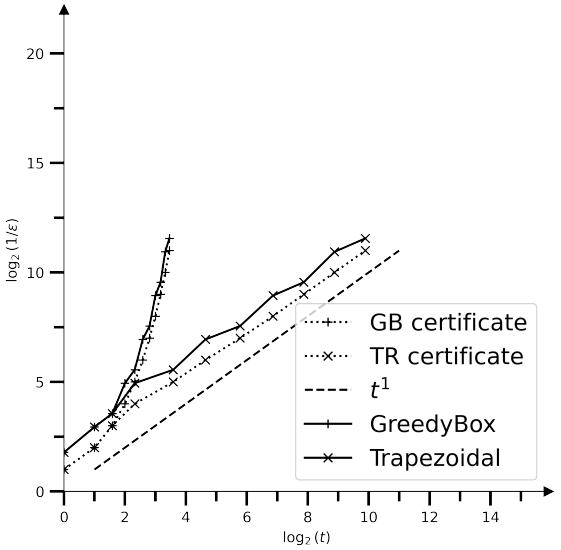
we can see that GreedyBox provides certification of ε-accuracy approximately 30 times as fast as the trapezoidal rule. For C^2 functions, this certificate will never be a good upper bound of the real error, that often is of order $\varepsilon^{-1/2}$ for all three algorithms.



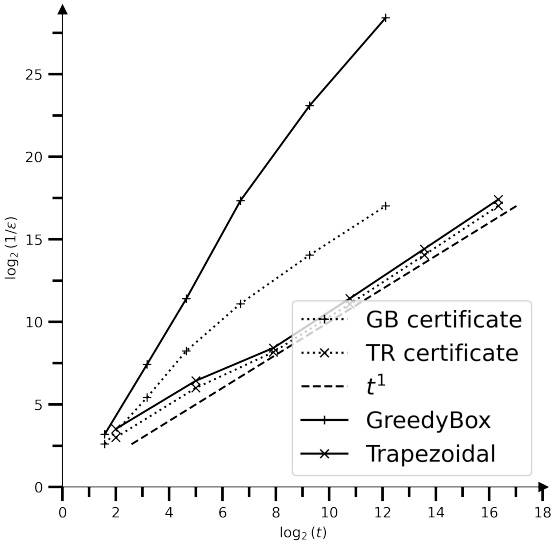
(a) Certificate and error on $f\colon x \mapsto x^2$



(b) Certificate and error on $f\colon x \mapsto x^{1/10}$



(c) Certificate and error on $f(x) = \mathbb{1}_{\{x \geqslant 0.3\}}$



(d) Certificate and error on g^t

Figure 5.5: Comparison of GreedyBox with the trapezoidal rule. Logarithmic scale of the inverse of the error w.r.t the number of evaluations.

5.6 Conclusion and future works

In this paper, we studied the problem of approximating a non-decreasing function f in $L^p(\mu)$ norm, with sequential (adaptive) access to its values. We first proved an f-dependent lower bound on the stopping time that holds for all algorithms with guaranteed $L^p(\mu)$ error after stopping. We then presented the GreedyBox algorithm (inspired from Novak (1992)) and showed that up to logarithmic factors, it is optimal among all such algorithms, for each non-decreasing function f. As a direct consequence, we showed for the integral

estimation problem that GreedyBox can be combined with additional randomization to get an improved rate in expectation. For the L^p-approximation problem, we also investigated to what extent the $L^p(\mu)$ error of GreedyBox can decrease faster (than guaranteed by the algorithm) for piecewise-C^2 functions. Put briefly, up to logarithmic factors, GreedyBox automatically achieves the improved (and optimal) rate of $\varepsilon^{-1+(\frac{1-\alpha}{1+p})_+}$ for a large number $\varepsilon^{-\alpha}$ of singularities, $\alpha \geqslant 1/2$, and a simple algorithmic variant (GreedyWidthBox) achieves this rate for any value of α. In particular, our results highlight multiple performance gaps between adaptive and non-adaptive algorithms, C^k and piecewise-C^k functions, as well as monotone or non-monotone functions. We also provided numerical experiments to illustrate our theoretical results.

Several interesting questions about GreedyBox are left open. First, similarly to the faster rates proved for piecewise-C^2 functions, it would be interesting to investigate optimal rates for the average $L^p(\mu)$ error, when f is drawn at random from a probability distribution over the set of monotone functions (as, e.g., in Novak (1992)). Second, for any fixed non-decreasing function f, it would be useful to derive the limit (if any) of the empirical distribution of the points queried by GreedyBox. When $p = 1$ and μ is the Lebesgue measure on $[0, 1]$, we conjecture that this limit exists and has a density roughly proportional to the square-root of the derivative of f almost everywhere. Solving this problem would help complete our understanding of the behavior of GreedyBox, and of how it precisely adapts to the complexity of any function f.

Several generalizations would also be worth investigating in the future. A seemingly straightforward direction is to work with Lipschitz functions on compact intervals; an algorithm defined similarly to GreedyBox but with parallelograms instead of rectangular boxes seems perfectly fit for this problem. Other natural directions consist in extending our f-dependent bounds to multivariate monotone functions (in the spirit of, e.g., Papageorgiou (1993)), to functions of known bounded variation, or variants of these function classes (e.g., entirely monotone functions, functions of bounded Hardy-Krause variation, see Fang et al. (2021)).

Another natural and interesting research avenue is to address the case of noisy evaluations of f. We believe that, under some known assumptions on the noise distribution, similar sample complexity guarantees (with slower rates) can be achieved by using a mini-batch variant of GreedyBox, which reduces the impact of noise by sampling multiple points and computing the average. Solving this problem would be a way to efficiently estimate cumulative distribution functions under a special censored feedback in the same spirit as in Abernethy et al. (2016).

Acknowledgements

The authors would like to thank Wouter Koolen for insightful discussions at the beginning of the work, as well as Peter Bartlett for fruitful feedback about improved rates for piecewise-regular functions. This work has benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French "Investing for the Future – PIA3" program under the Grant agreement ANR-19-P3IA-0004. Sébastien Gerchinovitz and Étienne de Montbrun gratefully acknowledge the support of IRT Saint Exupéry and the DEEL

project.[9]

5.A Elementary properties of box-covers

5.A.1 A general upper bound (proof of Lemma 5.1)

Lemma 5.1. For all non-decreasing functions $f : [0,1] \to [0,1]$ and $\varepsilon > 0$, the quantity $\mathcal{N}_p(f,\varepsilon)$ is well defined and upper bounded by

$$\mathcal{N}_p(f,\varepsilon) \leqslant \lceil 1/\varepsilon \rceil \,.$$

Proof. Let $n = \lceil 1/\varepsilon \rceil \geqslant 1$. In order to prove the lemma, we exhibit a sequence $\mathcal{B}$ of at most n boxes, show that it is a box-cover of f, and that the generalized areas of the boxes B are such that $(\sum_B \mathcal{A}_p(B)^p)^{1/p} \leqslant \varepsilon$.

For $i \in \{1,\dots,n\}$ let $x_i = \sup\{0 \leqslant x \leqslant 1 \colon f(x) \leqslant \frac{i}{n}\}$. We also set $x_0 = 0$. Note that $x_n = 1$. For $i \in \{1,\dots,n\}$, we define $B_i = [x_{i-1}, x_i] \times \left[\frac{i-1}{n}, \frac{i}{n}\right]$. Without loss of generality, we assume that $x_{i-1} < x_i$ for all $i \in \{1,\dots,n\}$. (Otherwise, we just remove all B_i's such that $x_{i-1} = x_i$, and remark below that the B_i's still cover the graph of f except maybe at the x_i's.)

We start by showing that $\mathcal{B} = (B_i)_{1 \leqslant i \leqslant n}$ is a box-cover of f. First, $B_1,\dots,B_n$ are adjacent boxes by construction. Note also that the graph of f is included in the union of the B_i's except maybe at the x_i's. Indeed, for all $i \in \{1,\dots,n\}$ and all $x \in (x_{i-1}, x_i)$, we have $\frac{i-1}{n} \leqslant f(x) \leqslant \frac{i}{n}$ by definition of x_{i-1} and x_i, and by monotonicity of f. Therefore, $(x, f(x)) \in B_i$. This proves that $\mathcal{B}$ is a box-cover of f.

We are left to show that $(\sum_{i=1}^n \mathcal{A}_p(B_i)^p)^{1/p} \leqslant \varepsilon$. We have

$$\sum_{i=1}^n \mathcal{A}_p(B_i)^p = \sum_{i=1}^n \left(\frac{i}{n} - \frac{i-1}{n}\right)^p \mu((x_{i-1}, x_i)) = \frac{1}{n^p} \sum_{i=1}^n \mu((x_{i-1}, x_i)) \leqslant \frac{1}{n^p} \,.$$

This entails that $(\sum_{i=1}^n \mathcal{A}_p(B_i)^p)^{1/p} \leqslant \frac{1}{n} \leqslant \varepsilon$, which concludes the proof. □

5.A.2 Technical lemmas on box-covers

We state here two elementary lemmas about box-covers. The first one below indicates that box-covers of f with desired properties exist not only for $n = \mathcal{N}_p(f,\varepsilon)$ (or $n = \mathcal{N}'_p(f,\varepsilon)$) but also for all larger values of n.

Lemma 5.14. *Let $\varepsilon > 0$ and $f : [0,1] \to [0,1]$ be any non-decreasing function. Then,*

- *for all $n \geqslant \mathcal{N}_p(f,\varepsilon)$, there exists a box-cover $B_1,\dots,B_n$ of f such that $\left(\sum_{i=1}^n \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon$;*
- *for all $n \geqslant \mathcal{N}'_p(f,\varepsilon)$, there exists a box-cover $B_1,\dots,B_n$ of f such that $\mathcal{A}_p(B_i) \leqslant \varepsilon$ for all $i = 1,\dots,n$.*

[9] https://www.deel.ai/

Proof. The proof of the two items is straightforward. For the first one, consider any box-cover $B_1, \dots, B_N$ of f, with $N = \mathcal{N}_p(f,\varepsilon)$, such that $\left(\sum_{i=1}^N \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon$. Then, split the last box B_N vertically into $n-N+1$ sub-boxes $B'_N, \dots, B'_n$. We can immediately see that

$$\left(\sum_{i=1}^{N-1} \mathcal{A}_p(B_i)^p + \sum_{i=N}^{n} \mathcal{A}_p(B'_i)^p\right)^{1/p} \leqslant \left(\sum_{i=1}^{n} \mathcal{A}_p(B_i)^p\right)^{1/p} \leqslant \varepsilon\,,$$

so that the sequence of boxes $B_1, \dots, B_{N-1}, B'_N, \dots, B'_n$ is a box-cover of f with cardinality n that satisfies the desired generalized area property. The second item can be proved similarly. □

The next simple lemma indicates that the values y_j^- and y_j^+ in any box-cover of f are necessary lower and upper bounds on the values of f inside the boxes.

Lemma 5.15. *Let $f : [0,1] \to [0,1]$ be non-decreasing, and $B_1, \dots, B_n$ be any box-cover of f, with $B_j = [c_{j-1}, c_j] \times [y_j^-, y_j^+]$ for all $j = 1, \dots, n$, for some sequence $0 = c_0 < \dots < c_n = 1$. Then, for all $j = 1, \dots, n$,*

$$y_j^- \leqslant \inf_{x > c_{j-1}} f(x) \qquad \text{and} \qquad y_j^+ \geqslant \sup_{x < c_j} f(x)\,.$$

Proof. Let $x \in (c_{j-1}, c_j)$. Since $(x, f(x)) \in \cup_{j'=1}^n B_{j'}$ and $x \notin [c_{k-1}, c_k]$ for any $k \neq j$, we have $(x, f(x)) \in B_j$ and therefore

$$y_j^- \leqslant f(x) \qquad \text{and} \qquad y_j^+ \geqslant f(x)\,.$$

We conclude by taking the infimum or supremum over all $x \in (c_{j-1}, c_j)$ and by using the fact that f is non-decreasing. □

5.A.3 Relationship between $\mathcal{N}_p(f,\varepsilon)$ and $\mathcal{N}'_p(f,\varepsilon)$

In this section, we compare the two complexity quantities $\mathcal{N}_p(f,\varepsilon)$ and $\mathcal{N}'_p(f,\varepsilon)$ defined in Section 5.1.2.

The quantity $\mathcal{N}'_p(f,\varepsilon)$ always satisfies $\mathcal{N}'_p(f,\varepsilon) \leqslant \mathcal{N}_p(f,\varepsilon)$. The next lemma relates the two complexity notions in a tighter way. Intuitively, if all the boxes B of a minimal box-cover of f satisfying $\sum_B \mathcal{A}_p(B)^p \leqslant \varepsilon^p$ had similar generalized areas $\mathcal{A}_p(B)$, these areas would be close to $\varepsilon/\mathcal{N}_p(f,\varepsilon)^{1/p}$, so that

$$\mathcal{N}'_p(f, \varepsilon/\mathcal{N}_p(f,\varepsilon)^{1/p}) \lesssim \mathcal{N}_p(f,\varepsilon).$$

The following lemma implies that indeed

$$\mathcal{N}'_p(f, \varepsilon/\mathcal{N}_p(f,\varepsilon)^{1/p}) \approx \mathcal{N}_p(f,\varepsilon)$$

up to a factor of 2.

Lemma 5.16. *Let $\varepsilon > 0$ and $n \geqslant \mathcal{N}_p(f,\varepsilon)$. Then,*

$$\mathcal{N}'_p\Big(f, \frac{\varepsilon}{n^{1/p}}\Big) \leqslant 2n \qquad \text{and} \qquad \mathcal{N}_p(f,\varepsilon) \leqslant \mathcal{N}'_p\Big(f, \frac{\varepsilon}{\mathcal{N}_p(f,\varepsilon)^{1/p}}\Big)\,.$$

In the proof of Theorem 5.3 we only use the first inequality $\mathcal{N}'_p(f, \frac{\varepsilon}{n^{1/p}}) \leqslant 2n$. The second inequality shows that this step is tight up to a constant of 2.

Proof. Let $f : [0,1] \to [0,1]$ be a non-decreasing function and $\varepsilon > 0$.

(a) We prove that $\mathcal{N}'_p(f, \varepsilon/n^{1/p}) \leqslant 2n$. For the sake of readability, we assume that μ is the Lebesgue measure on $[0,1]$, and will explain at the end how to adapt the proof for an arbitrary probability measure μ. By $n \geqslant \mathcal{N}_p(f,\varepsilon)$ and Lemma 5.14, there exists a box-cover $B_1, \dots, B_n$ of f satisfying $\sum_{i=1}^n \mathcal{A}_p(B_i)^p \leqslant \varepsilon^p$. Following a technique from Novak (1992, Proof of Theorem 3), we now divide the B_i's into as many sub-boxes as necessary so that each of their generalized areas is below $\varepsilon/n^{1/p}$. We will show that the overall number of resulting boxes is at most $2n$.

Let $i \in \{1, \dots, n\}$. We split the box $B_i = [x^-, x^+] \times [y^-, y^+]$ in a vertical fashion by splitting $[x^-, x^+]$ into $k_i = \lfloor n\mathcal{A}_p(B_i)^p/\varepsilon^p \rfloor + 1$ intervals of equal sizes $[x_0, x_1], \dots, [x_{k-1}, x_{k_i}]$, where $x_j = x^- + j(x^+ - x^+)/k_i$. The choice of k_i ensures that $\mathcal{A}_p(B_i)^p \leqslant k_i \varepsilon^p / n$.

Then, we define the smaller boxes $B_i^{(j)} := [x_{j-1}, x_j] \times [y^-, y^+]$ for $j = 1, \dots, k_i$. Their generalized areas are given by $\mathcal{A}_p(B_i^{(j)}) = ((x^+ - x^-)/k_i)^{1/p}(y^+ - y^-) = \mathcal{A}_p(B_i)/k_i^{1/p} \leqslant \varepsilon/n^{1/p}$ as required. Furthermore, the new total number of boxes is

$$\sum_{i=1}^n k_i = \sum_{i=1}^n \left(\left\lfloor \frac{n\mathcal{A}_p(B_i)^p}{\varepsilon^p} \right\rfloor + 1 \right) \leqslant n + \frac{n}{\varepsilon^p} \sum_{i=1}^n \mathcal{A}_p(B_i)^p \leqslant 2n\,.$$

Since $\cup_{i=1}^n \cup_{j=1}^{k_i} B_i^{(j)} = \cup_{i=1}^n B_i$ contains the graph of f except maybe at the B_i's endpoints (and thus everywhere outside the $B_i^{(j)}$'s endpoints), we have shown that $\mathcal{N}'_p(f, \varepsilon/n^{1/p}) \leqslant 2n$, as desired.

When μ is not the Lebesgue measure on $[0,1]$, the proof can be slightly adapted as follows. For each box $B_i = [x^-, x^+] \times [y^-, y^+]$ as above, we distinguish two cases.
Case 1: if $\mu((x^-, x^+)) = 0$, we set $k_i = 1$, $x_0 = x^-$, and $x_1 = x^+$ as above.
Case 2: if $\mu((x^-, x^+)) > 0$, we set $k_i = \lfloor n\mathcal{A}_p(B_i)^p/\varepsilon^p \rfloor + 1$, $x_0 = x^-$, and $x_{k_i} = x^+$ as above. However, for any $1 \leqslant j < k_i$ (if any), we take x_j as a quantile of order j/k_i of the conditional distribution $\mu(\cdot | (x^-, x^+))$. Then, among all sub-boxes $B_i^{(j)} := [x_{j-1}, x_j] \times [y^-, y^+]$, $j = 1, \dots, k_i$, we only keep the non-degenerate ones, i.e., those such that $x_{j-1} < x_j$.

The rest of the proof remains unchanged. In particular, each remaining sub-box satisfies $\mathcal{A}_p(B_i^{(j)}) \leqslant \varepsilon/n^{1/p}$, and the total number of sub-boxes is at most of $\sum_{i=1}^n k_i \leqslant 2n$.

(b) Before proving the second inequality, we prove an intermediate result: for all $n_1 \geqslant \mathcal{N}'_p(f,\varepsilon)$, we have

$$\mathcal{N}_p(f, n_1^{1/p}\varepsilon) \leqslant n_1\,. \tag{5.9}$$

Since $n_1 \geqslant \mathcal{N}'_p(f,\varepsilon)$, we can consider a box-cover $B_1, \dots, B_{n_1}$ of f with generalized areas at most of ε each. The result immediately follows from

$$\Big(\sum_{i=1}^{n_1} \mathcal{A}_p(B_i)^p \Big)^{1/p} \leqslant n_1^{1/p}\varepsilon\,.$$

(c) We write $n_\varepsilon = \mathcal{N}_p(f,\varepsilon)$ for simplicity. We prove that $n_\varepsilon \leqslant \mathcal{N}'_p(f,\varepsilon/n_\varepsilon^{1/p})$ by contradiction. Let us assume that $\mathcal{N}'_p(f,\varepsilon/n_\varepsilon^{1/p}) < n_\varepsilon$ and define

$$\varepsilon' := \varepsilon \cdot \left(\frac{\mathcal{N}'_p(f,\varepsilon/n_\varepsilon^{1/p})}{n_\varepsilon}\right)^{1/p} < \varepsilon\,.$$

Then using Inequality (5.9), with ε substituted with $\varepsilon/n_\varepsilon^{1/p}$, we get

$$\mathcal{N}_p(f,\varepsilon) \overset{\varepsilon'<\varepsilon}{\leqslant} \mathcal{N}_p(f,\varepsilon') = \mathcal{N}_p\Big(f, \mathcal{N}'_p\Big(f, \frac{\varepsilon}{n_\varepsilon^{1/p}}\Big)^{1/p} \frac{\varepsilon}{n_\varepsilon^{1/p}}\Big) \overset{(5.9)}{\leqslant} \mathcal{N}'_p\Big(f, \frac{\varepsilon}{n_\varepsilon^{1/p}}\Big)\,,$$

which contradicts the assumption and thus proves the desired inequality.

□

5.B Omitted proofs

5.B.1 Proof of Lemma 5.4

Lemma 5.4. Let $f : [0,1] \to [0,1]$ be non-decreasing, $p \geqslant 1$ and $\varepsilon \in (0,1]$. For any $t \in \{1,\dots,\tau_\varepsilon\}$,

$$\|\widehat{f}_t - f\|_p^p := \int_0^1 \left|\widehat{f}_t(x) - f(x)\right|^p \mathrm{d}\mu(x) \leqslant \sum_{k=1}^{t} (a_k^t)^p =: \xi_t\,.$$

Proof. Remark that the approximation $\widehat{f}_t$ is the continuous piecewise-affine function such that $\widehat{f}_t(b_k^t) = f(b_k^t)$ for all $k \in \{0,\dots,t\}$. Therefore, it suffices to show that for each $k = 1,\dots,t$,

$$\int_{(b_{k-1}^t, b_k^t)} \left|\widehat{f}_t(x) - f(x)\right|^p \mathrm{d}\mu(x) \leqslant (a_k^t)^p\,,$$

which follows easily from the fact that $\left|\widehat{f}_t(x) - f(x)\right| \leqslant f(b_k^t) - f(b_{k-1}^t)$ on the k-th box (since f is non-decreasing), and by definition of $a_k^t := (\mu(b_{k-1}^t, b_k^t))^{1/p}(f(b_k^t) - f(b_{k-1}^t))$. Summing over $k = 1,\dots,t$ concludes the proof. □

As a supplement, we are left with proving the special case of the Lebesgue measure, for which the result holds with a multiplicative factor of $1/(p+1)$.

Lemma 5.17. *Let $f : [0,1] \to [0,1]$ be non-decreasing. At any round $t \geqslant 1$,*

$$\|\widehat{f}_t - f\|_p^p := \int_0^1 (\widehat{f}_t(x) - f(x))^p \,\mathrm{d}x \leqslant \frac{1}{1+p} \sum_{k=1}^{t} (a_k^t)^p\,.$$

Proof. Summing over $k = 1,\dots,t$ it suffices to show that for each $k = 1,\dots,t$,

$$\int_{b_{k-1}^t}^{b_k^t} \left|\widehat{f}_t(x) - f(x)\right|^p \mathrm{d}x \leqslant \frac{(a_k^t)^p}{1+p}\,.$$

Let $k \in \{1, \dots, t\}$. To ease the notation, we make a change of variables and define for all $u \in [0,1]$:

$$g(u) := \frac{f\big((b_k^t - b_{k-1}^t)u + b_{k-1}^t\big) - f(b_{k-1}^t)}{f(b_k^t) - f(b_{k-1}^t)}\,.$$

We have

$$\begin{aligned}\int_{b_{k-1}^t}^{b_k^t} \left|\widehat{f}_t(x) - f(x)\right|^p \mathrm{d}x &= \big(f(b_k^t) - f(b_{k-1}^t)\big)^p (b_k^t - b_{k-1}^t) \int_0^1 |u - g(u)|^p \,\mathrm{d}u \\ &= (a_k^t)^p \int_0^1 |u - g(u)|^p \,\mathrm{d}u\,.\end{aligned}$$

The function g is non-decreasing over $[0,1]$ with $g(0) = 0$ and $g(1) = 1$. It remains to control $\int_0^1 |u - g(u)|^p \, du$ by $1/(1+p)$. which is done in the following.

First we remark that we can assume $g(1/2) \neq 1/2$. Otherwise, since g is non-decreasing $|g(u) - u| \in [0, 1/2]$ for all $u \in [0,1]$ and $\int_0^1 |u - g(u)|^p \, du \leqslant 2^{-p} \leqslant (1+p)^{-1}$, which concludes.

We consider the two points $u_- \leqslant 1/2 \leqslant u_+$ such that the sign of $g(x) - x$ does not change over (u_-, u_+). More formally, they are defined as:

$$u_- = \inf\Big\{u \in [0, 1/2] : \quad \forall x \in (u, 1/2] \quad \big(g(1/2) - 1/2\big)\big(g(x) - x\big) > 0\Big\}$$

and

$$u_+ = \sup\Big\{u \in [1/2, 1] : \quad \forall x \in [1/2, u) \quad \big(g(1/2) - 1/2\big)\big(g(x) - x\big) > 0\Big\}\,.$$

Note that the sign of $g(x) - x$ is constant over $\{1/2\} \cup (u_-, u_+)$ since $g(1/2) \neq 1/2$. Now, we show the following two facts

$$\forall u < u_-, \quad g(u) \leqslant u_- \qquad \text{and} \qquad \forall u > u_+, \quad g(u) \geqslant u_+\,. \tag{5.10}$$

Indeed, let $u < u_-$, by definition of u_-, there exists $x \in [u, u_-]$ such that

$$\big(g(1/2) - 1/2\big)\big(g(x) - x\big) \leqslant 0\,,$$

and thus using again the definition of u_- for all $x' \in (u_-, 1/2]$, $\big(g(x') - x'\big)\big(g(x) - x\big) \leqslant 0$. If $g(x) \leqslant x$, we are done since $g(u) \leqslant g(x) \leqslant x \leqslant u_-$ because g is non-decreasing and $u \leqslant x \leqslant u_-$. Otherwise using $u \leqslant x'$, $g(u) \leqslant g(x') \leqslant x'$ and making $x' \to u_-$ concludes the first inequality of (5.10). The second inequality can be proved similarly.

Therefore from (5.10), for all $u \leqslant u_-$, $|g(u) - u| \leqslant u_-$ and all $u \geqslant u_+$, $|g(u) - u| \leqslant 1 - u_+$ which yields

$$\int_0^{u_-} |g(u) - u|^p \,\mathrm{d}u \leqslant u_-^{1+p} \quad \text{and} \quad \int_{u_+}^1 |g(u) - u|^p \,\mathrm{d}u \leqslant (1 - u_+)^{1+p}\,. \tag{5.11}$$

Furthermore, $g(u) - u$ does not change sign over (u_-, u_+), which entails:

$$\int_{u_-}^{u_+} |g(u) - u|^p \,\mathrm{d}u \leqslant \max\left\{\int_{u_-}^{u_+} (u_+ - u)^p \,\mathrm{d}u, \int_{u_-}^{u_+} (u - u_-)^p \,\mathrm{d}u\right\} = \frac{(u_+ - u_-)^{p+1}}{p+1}\,. \tag{5.12}$$

Summing Inequalities (5.11) and (5.12), we get

$$\int_0^1 |g(u)-u|^p \,\mathrm{d}u \leqslant u_-^{1+p} + (1-u_+)^{1+p} + \frac{(u_+-u_-)^{1+p}}{1+p}$$
$$\leqslant \sup_{0\leqslant u_-\leqslant 1/2\leqslant u_+\leqslant 1} \left\{ u_-^{1+p} + (1-u_+)^{1+p} + \frac{(u_+-u_-)^{1+p}}{1+p} \right\} = \frac{1}{1+p}\,.$$

The supremum is reached for $(u_-,u_+)=(0,1)$. □

5.B.2 Proof of Lemma 5.5

Lemma 5.5. Let $f:[0,1]\to[0,1]$ be non-decreasing, $p\geqslant 1$ and $\varepsilon\in(0,1]$. Define $\tau'_\varepsilon := 2(1+\lceil p\log_2(1/\varepsilon)\rceil)\mathcal{N}'_p(f,\varepsilon)$, and assume that GreedyBox is such that $\tau_\varepsilon > \tau'_\varepsilon$. Then, at time τ'_ε, all the boxes maintained by GreedyBox have a generalized area bounded from above by ε, i.e., $a_k^{\tau'_\varepsilon}\leqslant\varepsilon$ for all $k\in\{1,\dots,\tau'_\varepsilon\}$.

Proof. By definition of $m:=\mathcal{N}'_p(f,\varepsilon)$, we can fix a box-cover $B_1,\dots,B_m$ of f such that $\mathcal{A}_p(B_j)\leqslant\varepsilon$ for all j. Recall that these boxes are adjacent, i.e., they are of the form $B_j=[c_{j-1},c_j]\times[y_j^-,y_j^+]$ for some nodes $0=c_0<\ldots<c_m=1$. The inequalities $\mathcal{A}_p(B_j)\leqslant\varepsilon$ thus translate into $(y_j^+-y_j^-)\mu((c_{j-1},c_j))^{1/p}\leqslant\varepsilon$ for all j.

In this proof we will compare the boxes maintained by GreedyBox with the boxes B_j above. We first need the following definition: at each round $t\geqslant 1$, we say that a box $B=[b_{k-1}^t,b_k^t]\times[f(b_{k-1}^t),f(b_k^t)]$, $1\leqslant k\leqslant t$, maintained by GreedyBox is

- *internal* if $[b_{k-1}^t,b_k^t]\subset(c_{j-1},c_j)$ for some $j=1,\dots,m$;
- *overlapping* if $[b_{k-1}^t,b_k^t]\ni c_j$ for some $j=1,\dots,m$.

Note that exactly one of the two cases above must hold true. In the sequel, we denote by $\widehat{B}_t=[b_{k_*^t-1}^t,b_{k_*^t}^t]\times[f(b_{k_*^t-1}^t),f(b_{k_*^t}^t)]$ the box selected by GreedyBox at time t. Since $\widehat{B}_t$ is necessary of one of the two types above, we can distinguish between the following two cases.

Case 1: there exists $t\in\{1,2,\dots,\tau'_\varepsilon\}$ such that $\widehat{B}_t$ is internal. In this case, letting $j=1,\dots,m$ be the corresponding index, we have, by Lemma 5.15 in Appendix 5.A.2,

$$c_{j-1}<b_{k_*^t-1}^t<b_{k_*^t}^t<c_j \qquad\text{and}\qquad y_j^-\leqslant f(b_{k_*^t-1}^t)\leqslant f(b_{k_*^t}^t)\leqslant y_j^+\,.$$

Therefore, the generalized areas of $\widehat{B}_t$ and B_j satisfy $a_{k_*^t}^t=\mathcal{A}_p(\widehat{B}_t)\leqslant\mathcal{A}_p(B_j)\leqslant\varepsilon$ by construction of B_j. Now, by definition of k_*^t in Algorithm 13:

$$\max_{1\leqslant k\leqslant t} a_k^t = a_{k_*^t}^t\leqslant\varepsilon\,.$$

This concludes the proof of the lemma in this case, since the quantity $\max_{1\leqslant k\leqslant t}a_k^t$ can only decrease between rounds t and τ'_ε.

Case 2: $\widehat{B}_t$ is an overlapping box at every round $t\in\{1,2,\dots,\tau'_\varepsilon\}$. In this case, we say that all nodes c_j lying in $[b_{k_*^t-1}^t,b_{k_*^t}^t]$ are *activated* at time t. More precisely, we say that

a node c_j is *left-activated* when $c_j \in (b^t_{k^t_*-1}, b^t_{k^t_*}]$ (part of the box $\widehat{B}_t$ lies on the left of c_j), and that it is *right-activated* when $c_j \in [b^t_{k^t_*-1}, b^t_{k^t_*})$ (part of the box $\widehat{B}_t$ lies on the right of c_j).

Since $b^t_{k^t_*-1} < b^t_{k^t_*}$ by construction, at least one node c_j is either left-activated or right-activated at every round $t \in \{1, 2, \dots, \tau'_\varepsilon\}$, so that

$$\sum_{t=1}^{\tau'_\varepsilon} \underbrace{\sum_{j=1}^{m} \sum_{\sigma \in \{\text{left,right}\}} \mathbb{1}_{\{c_j \text{ is } \sigma\text{-activated at round } t\}}}_{\geqslant 1} \geqslant \tau'_\varepsilon \,.$$

Inverting sums and recognizing $N_{j,\sigma} = \sum_{t=1}^{\tau'_\varepsilon} \mathbb{1}_{\{c_j \text{ is } \sigma\text{-activated at round } t\}}$ to be the number of rounds when node c_j is σ-activated, we can see that

$$\sum_{j=1}^{m} \sum_{\sigma \in \{\text{left,right}\}} N_{j,\sigma} \geqslant \tau'_\varepsilon \,,$$

so that

$$\max_{1 \leqslant j \leqslant m} \max_{\sigma \in \{\text{left,right}\}} N_{j,\sigma} \geqslant \frac{\tau'_\varepsilon}{2m} \,.$$

Combining the last inequality with the definition of τ'_ε and $m := \mathcal{N}'_p(f, \varepsilon)$, we obtain the following intermediate result.

Remark 5.18. There exists $j \in \{1, \dots, \mathcal{N}'_p(f,\varepsilon)\}$ and $\sigma \in \{\text{left}, \text{right}\}$ such that the node c_j is σ-activated at least

$$\frac{\tau'_\varepsilon}{2\mathcal{N}'_p(f,\varepsilon)} = 1 + \lceil p \log_2(1/\varepsilon) \rceil$$

times within the set of rounds $\{1, \dots, \tau'_\varepsilon\}$.

We now focus on a single pair (j, σ) provided by Remark 5.18 above. We follow the evolution of the box $\widetilde{B}_t$ maintained by GreedyBox that lies on the σ-side of c_j. More formally, if we write $\widetilde{B}_t = [b^t_{k-1}, b^t_k] \times [f(b^t_{k-1}), f(b^t_k)]$, this means that $c_j \in (b^t_{k-1}, b^t_k]$ if $\sigma = \text{left}$, and that $c_j \in [b^t_{k-1}, b^t_k)$ if $\sigma = \text{right}$. Note that, at any round t, such a box $\widetilde{B}_t$ indeed exists and is unique.

Note that, at all rounds $t \in \{1, \dots, \tau'_\varepsilon\}$ when c_j is σ-activated, we have $\widetilde{B}_t = \widehat{B}_t$, so that the box $\widetilde{B}_t$ is replaced (see Step 4 in Algorithm 13) by two boxes whose generalized widths are at most half that of $\widetilde{B}_t$. This is because x_{t+1} is a median of the conditional distribution $\mu(\cdot | (b^t_{k^t_*-1}, b^t_{k^t_*}))$. Since $\widetilde{B}_{t+1}$ is among these two boxes, we thus have

$$\operatorname{width}(\widetilde{B}_{t+1}) \leqslant \operatorname{width}(\widetilde{B}_t)/2$$

at each round $t \in \{1, \dots, \tau'_\varepsilon\}$ when c_j is σ-activated. Note also that $\widetilde{B}_{t+1} = \widetilde{B}_t$ at all other rounds $t \in \{1, \dots, \tau'_\varepsilon\}$; at such rounds, $\operatorname{width}(\widetilde{B}_{t+1}) = \operatorname{width}(\widetilde{B}_t)$. Combining the last two properties, and denoting by τ the round when c_j is σ-activated for the $\lceil p \log_2(1/\varepsilon) \rceil$-th time, we get

$$\operatorname{width}(\widetilde{B}_{\tau+1}) \leqslant 2^{-\lceil p \log_2(1/\varepsilon) \rceil} \leqslant \varepsilon^p \,.$$

To conclude, denote by τ' the round when c_j is σ-activated for the $(\lceil p\log_2(1/\varepsilon)\rceil+1)$-th time. Note that $\widehat{B}_{\tau'} = \widetilde{B}_{\tau'}$ so that

$$\mathrm{width}(\widehat{B}_{\tau'}) = \mathrm{width}(\widetilde{B}_{\tau'}) \leqslant \mathrm{width}(\widetilde{B}_{\tau+1}) \leqslant \varepsilon^p\,,$$

where the first inequality follows from $\tau' \geqslant \tau+1$ and the fact that $\mathrm{width}(\widetilde{B}_t)$ is non-increasing over time. Therefore, by definition of $\widehat{B}_{\tau'}$, all boxes maintained by GreedyBox at time τ' have generalized areas bounded by

$$\max_{1\leqslant k\leqslant \tau'} a_k^{\tau'} \leqslant \mathcal{A}_p(\widehat{B}_{\tau'}) \leqslant (1-0)\times \mathrm{width}(\widehat{B}_{\tau'})^{1/p} \leqslant \varepsilon\,.$$

We conclude the proof by noting that $\tau' \leqslant \tau'_\varepsilon$, so that $\max_{1\leqslant k\leqslant \tau'_\varepsilon} a_k^{\tau'_\varepsilon} \leqslant \max_{1\leqslant k\leqslant \tau'} a_k^{\tau'} \leqslant \varepsilon$. □

5.B.3 Proof of Lemma 5.6

Lemma 5.6. Let $f:[0,1]\to[0,1]$ be non-decreasing, $p\geqslant 1$ and $\varepsilon\in(0,1]$. For any $t\in\{1,\dots,\lfloor\tau_\varepsilon/2\rfloor\}$, we have $\xi_{2t}\leqslant \xi_t/2$. Therefore, for all $t\leqslant s$ in $\{1,\dots,\tau_\varepsilon\}$,

$$\xi_s \leqslant \frac{\xi_t}{2^{\lfloor\log_2(s/t)\rfloor}} \leqslant \Big(\frac{2t}{s}\Big)\xi_t\,.$$

Proof. We first prove that $\xi_{2t}\leqslant\xi_t/2$ for all $t\in\{1,\dots,\lfloor\tau_\varepsilon/2\rfloor\}$. To do so, recall that at any round $1\leqslant t'\leqslant\tau_\varepsilon$, GreedyBox maintains t' boxes given by $[b^{t'}_{k-1},b^{t'}_k]\times[f(b^{t'}_{k-1}),f(b^{t'}_k)]$, $k=1,\dots,t'$. We write $a^{t'}_{(1)}\geqslant a^{t'}_{(2)}\geqslant\dots\geqslant a^{t'}_{(t')}$ for their generalized areas $a^{t'}_k$ sorted in decreasing order.

Part 1: We first show by induction on $k=1,\dots,t$ that

(i) the box selected by GreedyBox at round $t-1+k$ (Step 2 in Algorithm 13) has a generalized area $a^{t-1+k}_{k^{t-1+k}_*}$ larger than or equal to $a^t_{(k)}$;

(ii) at least $t-k$ boxes of round $t+k$ are identical to boxes of round t.

Both (i) and (ii) are straightforward for $k=1$. Assume they are true for some $k\in\{1,\dots,t-1\}$. Next we prove (i) and (ii) with the index value $k+1$. At round $t+k$, the generalized area of the box selected by GreedyBox must be at least as large as the maximum generalized area of the $t-k$ boxes that are identical to boxes of round t (by (ii) with k). Therefore, this generalized area is larger than or equal to $a^t_{(k+1)}$, which proves (i). Property (ii) is immediate since only one box is selected at every round. This completes the induction.

Part 2: Note that, at each round $t-1+k\in\{t,t+1,\dots,2t-1\}$, the box $\widehat{B}_{t-1+k}$ selected by GreedyBox is replaced with two smaller boxes B' and B'' whose generalized areas satisfy

$$\mathcal{A}_p(B')^p+\mathcal{A}_p(B'')^p \leqslant \mathcal{A}_p(\widehat{B}_{t-1+k})^p/2\,,$$

since x_{t+1} is a median of $\mu(\cdot|(b^t_{k^t_*-1},b^t_{k^t_*}))$ and $(\delta')^p+(\delta'')^p\leqslant(\delta'+\delta'')^p$ for any $\delta',\delta''\geqslant 0$. Therefore, and by Property (i) above, at least $\mathcal{A}_p(\widehat{B}_{t-1+k})^p/2\geqslant(a^t_{(k)})^p/2$ is lost when summing the generalized areas to the power p at round $t+k$, compared to round $t+k-1$.

Therefore, the certificate ξ_{t+k} of the box-cover at the next round satisfies $\xi_{t+k} \leqslant \xi_{t-1+k} - (a^t_{(k)})^p/2$. Summing over $k = 1, \dots, t$ we get:

$$\xi_{2t} \leqslant \xi_t - \sum_{k=1}^{t} \frac{(a^t_{(k)})^p}{2} = \frac{\xi_t}{2}\,.$$

To see why this implies (5.5), it suffices to note that $s \geqslant \widetilde{s} := 2^{\lfloor \log_2(s/t) \rfloor} \cdot t$, so that

$$\xi_s \leqslant \xi_{\widetilde{s}} \leqslant \frac{\xi_t}{2^{\lfloor \log_2(s/t) \rfloor}}\,,$$

where the first inequality is because the certificate ξ_s can only decrease over time, and where the last inequality follows from the property $\xi_{2t} \leqslant \xi_t/2$ shown above. This concludes the proof. □

5.B.4 Proof of Theorem 5.10

Theorem 5.10. Let $f : [0,1] \to [0,1]$ be non-decreasing which satisfies Assumption 6 for some $C, \alpha > 0$. Let $\varepsilon > 0$, then Algorithm 14 satisfies

$$\mathbb{E}\Big[\big|\widehat{I}_{\tau_\varepsilon}(f) - I(f)\big|\Big] \leqslant \varepsilon.$$

Besides the number of function evaluations is bounded from above by

$$\tau_\varepsilon = \mathcal{O}\big(\log(1/\varepsilon)^{3/2} \varepsilon^{-\frac{1}{1/\alpha+1/2}}\big)\,.$$

Proof. Let $\varepsilon > 0$. First, we remark that the points $x_0, \dots, x_t$ defined by the deterministic version GreedyBox (defined in Algorithm 13) and the stochastic version (Algorithm 14) are identical. Only the stopping criterion and definition of $\widehat{I}_{\tau_\varepsilon}(f)$ differs from Algorithm 13. Therefore, we can apply Lemma 5.5.

Let $\varepsilon_1 \geqslant \varepsilon$ that will be fixed later as a function of ε. Denote $n_{\varepsilon_1} = C\varepsilon_1^{-\alpha} \geqslant \mathcal{N}(f, \varepsilon_1)$ by Assumption 6. Applying Lemma 5.5 to $\varepsilon_2 = \varepsilon_1/n_{\varepsilon_1} = \varepsilon_1^{1+\alpha}/C$, we get the following result. At time

$$\tau'_{\varepsilon_2} := (1 + \lceil \log_2(1/\varepsilon_2) \rceil)\mathcal{N}'(f, \varepsilon_2)$$

all the boxes maintained by Algorithm 14 have an area bounded from above by ε_2, i.e.,

$$a_k^{\tau'_{\varepsilon_2}} \leqslant \varepsilon_2, \qquad \forall k = 1, \dots, \tau'_{\varepsilon_2}\,.$$

From Lemma 5.16 since $n_{\varepsilon_1} \geqslant \mathcal{N}(f, \varepsilon_1)$ by Assumption 6,

$$\mathcal{N}'(f, \varepsilon_2) = \mathcal{N}'\Big(f, \frac{\varepsilon_1}{n_{\varepsilon_1}}\Big) \leqslant 2n_{\varepsilon_1} = 2C\varepsilon_1^{-\alpha}\,,$$

which substituted into the definition of τ'_{ε_2} yields

$$\tau'_{\varepsilon_2} \leqslant 2C(1 + \lceil \log_2(1/\varepsilon_2) \rceil)\varepsilon_1^{-\alpha}\,.$$

Thus, the certificate up to time τ'_{ε_2} is bounded from above as

$$\xi_{\tau_{\varepsilon_2}} = \frac{1}{2}\sum_{k=1}^{\tau'_{\varepsilon_2}}(a_k^{\tau'_{\varepsilon_2}})^2 \leqslant \frac{\tau'_{\varepsilon_2}(\varepsilon_2)^2}{2} \leqslant \frac{1}{C}(1+\lceil\log_2(1/\varepsilon_2)\rceil)\varepsilon_1^{2+\alpha}.$$

Now, to get rid of the multiplicative term, similarly to Theorem 5.3, we can apply Lemma 5.6 (which also works for StochasticGreedyBox) to replace $\xi_{\tau'_{\varepsilon_2}}$ with ξ_s for $s \geqslant \tau'_{\varepsilon_2}$. The choice $s = \tau'_{\varepsilon_2}(\frac{1}{C}(1+\lceil\log_2(1/\varepsilon_2)\rceil))^{1/2}$ yields

$$\xi_s \leqslant \varepsilon_1^{2+\alpha}.$$

Then, choosing $\varepsilon_1 = \varepsilon^{\frac{2}{2+\alpha}}$ implies $\xi_s \leqslant \varepsilon^2$. Therefore by definition of the stopping criterion

$$\tau_\varepsilon \leqslant s = \Big(\frac{1}{C}(1+\lceil\log_2(1/\varepsilon_2)\rceil)\Big)^{3/2}\varepsilon_1^{-\alpha} = \widetilde{\mathcal{O}}(\varepsilon^{-\frac{1}{1/\alpha+1/2}}).$$

This concludes the proof. □

5.B.5 Proof of Theorem 5.11

Theorem 5.11. Let $\alpha > 0$ and $\varepsilon \in (0,1]$. Let $f : [0,1] \to [0,1]$ be a non-decreasing and piecewise-C^2 function with a number of C^1-singularities bounded by $\varepsilon^{-\alpha}$ and such that $|f''(x)| \leqslant 1$ whenever it is defined. Then, there exists

$$t_\varepsilon = \begin{cases} \widetilde{\mathcal{O}}\Big(\varepsilon^{-1+\frac{1}{2p+2}}\Big) & \text{if } \alpha \leqslant \frac{1}{2} \\ \widetilde{\mathcal{O}}\Big(\varepsilon^{-1+\left(\frac{1-\alpha}{1+p}\right)_+}\Big) & \text{if } \alpha \geqslant \frac{1}{2} \end{cases}$$

such that $\|\widehat{f}_t - f\|_p \leqslant \varepsilon$ for all $t \geqslant t_\varepsilon$, where $\widehat{f}_t$ is the approximation of f returned by GreedyBox after t rounds.

Proof. Let $c \in (0,1]$, and $\gamma > 0$ be two constants to be fixed later by the analysis and set $\varepsilon' = c\varepsilon^\gamma$.

Step 1. We will now establish an upper bound on the number of evaluations, denoted by τ_ε, required by GreedyBox to ensure that all individual areas are smaller than ε'. From Lemma 5.5, we know that

$$\tau_\varepsilon \leqslant 2(1+\lceil p\log_2(1/\varepsilon')\rceil)\mathcal{N}'_1(f,\varepsilon'), \tag{5.13}$$

which can be further bounded from above by the use of Lemma 5.16 in the appendix with the choice $n := c^{-p}\lceil\varepsilon^{-\frac{\gamma p}{p+1}}\rceil$. Indeed, since $c \leqslant 1$ and by Lemma 5.1, $n \geqslant \lceil\varepsilon^{-\frac{\gamma p}{p+1}}\rceil \geqslant \mathcal{N}_p(f,\varepsilon^{\frac{\gamma p}{p+1}})$. Thus, Lemma 5.16 entails

$$\mathcal{N}'_1(f,\varepsilon') = \mathcal{N}'_1(f,c\varepsilon^\gamma) = \mathcal{N}'_1\Big(f, \frac{c\varepsilon^{\frac{\gamma p}{p+1}}}{\varepsilon^{-\frac{\gamma}{p+1}}}\Big) \leqslant \mathcal{N}'_1\Big(f, \frac{\varepsilon^{\frac{\gamma p}{p+1}}}{n^{1/p}}\Big) \leqslant 2n = 2c^{-p}\lceil\varepsilon^{-\frac{\gamma p}{p+1}}\rceil,$$

Therefore, plugging back into Inequality (5.13), the number of required evaluations is bounded as

$$\tau_\varepsilon \leqslant 4(1+\lceil p\log_2(1/\varepsilon')\rceil)c^{-p}\lceil\varepsilon^{-\frac{\gamma p}{p+1}}\rceil = \widetilde{\mathcal{O}}(c^{-p}\varepsilon^{-\frac{\gamma p}{p+1}}). \tag{5.14}$$

Step 2. We will now fix the values of $c \in (0,1]$ and $\gamma > 0$ in a way that ensures an approximation error in the L^p-norm smaller than ε.

For $1 \leqslant i \leqslant t$, let us denote by $B_i = [x_i, x_{i+1}] \times [f(x_i), f(x_{i+1})]$ the i^{th} box maintained by GreedyBox, define $w_i := x_{i+1} - x_i$ to be the width of the ith box and let $\widehat{f}_t$ be the piecewise-linear function returned by GreedyBox after t epochs. By the first step of this proof, for all $1 \leqslant i \leqslant t$,

$$\mathcal{A}_p(B_i) := \Big((f(x_{i+1}) - f(x_i))^p (x_{i+1} - x_i) \Big)^{1/p} \leqslant \varepsilon' ,$$

which implies by construction of $\widehat{f}_t$ (see Proof of Lemma 5.4), for all $1 \leqslant i \leqslant t$

$$\int_{x_i}^{x_{i+1}} |\widehat{f}_t(x) - f(x)|^p \, \mathrm{d}x \leqslant (\varepsilon')^p . \tag{5.15}$$

The remaining part of the proof revolves around using the above upper bound for the non-smooth pieces and a more refined upper bound for the C^1 pieces. Denote by $\mathcal{J} \subseteq \{1, \dots, t\}$ the indices of all boxes such that f is C^1 over $[x_i, x_{i+1}]$. Because f is piecewise-C^1 with at most K pieces, $\mathrm{Card}(\mathcal{J}^c) \leqslant K$. Therefore, the L^p error after t evaluations may be decomposed as

$$\begin{aligned} \|\widehat{f}_t - f\|_p^p = \int_0^1 |\widehat{f}_t(x) - f(x)|^p \, \mathrm{d}x &= \sum_{i=1}^{t} \int_{x_i}^{x_{i+1}} |\widehat{f}_t(x) - f(x)|^p \, \mathrm{d}x \\ &\overset{(5.15)}{\leqslant} \min\{K, t\} (\varepsilon')^p + \sum_{i \in \mathcal{J}} \int_{x_i}^{x_{i+1}} |\widehat{f}_t(x) - f(x)|^p \, \mathrm{d}x \end{aligned} \tag{5.16}$$

We are now left with bounding from above the L^p errors on the right-hand-side for all $i \in \mathcal{J}$. On one side, the bound (5.15) is valid for all $i \in \mathcal{J}$, which we bound further by $(\varepsilon')^p$. On the other side, we can use Lemma 5.20 that bounds the L^p approximation error of $\widehat{f}_t$ for any C^1 and piecewise-C^2 function to obtain

$$\int_{x_i}^{x_{i+1}} |\widehat{f}_t(x) - f(x)|^p \, \mathrm{d}x \leqslant M^p w_i^{2p+1} , \tag{5.17}$$

where $M \geqslant \frac{3}{2} \sup_{x \notin \mathcal{X}_2} |f''(x)|$, where $\mathcal{X}_2$ denotes the set of C^2-singularities. This prompt us to introduce the following function ϕ, that depends on the width of the intervals of $\mathcal{J}$:

$$\phi\left((w_i)_{i \in \mathcal{J}}\right) = \sum_{i \in \mathcal{J}} \min \left\{ (\varepsilon')^p, M^p w_i^{2p+1} \right\} .$$

From (5.16), the total error is thus bounded from above by

$$\|\widehat{f}_t - f\|_p^p \leqslant \min\{K, t\} (\varepsilon')^p + \phi\left((w_i)_{i \in \mathcal{J}}\right) . \tag{5.18}$$

It now remains to bound the function ϕ for any set $(w_i)_{i \in \mathcal{J}}$ of possible widths that could arise from GreedyBox. We thus need to solve the maximization problem

$$\max_{(w_i)_{i \in \mathcal{J}}} \phi((w_i)_{i \in \mathcal{J}}) \text{ such that } \sum_{i \in \mathcal{J}} w_i \leqslant 1 .$$

ϕ may be re-written in two terms:

$$\phi((w_i)_{i\in\mathcal{J}}) = \sum_{i\in\mathcal{J},\varepsilon'\leqslant M w_i^{2+\frac{1}{p}}} (\varepsilon')^p + \sum_{i\in\mathcal{J},\varepsilon'>M w_i^{2+\frac{1}{p}}} M^p w_i^{2p+1} . \tag{5.19}$$

Let us first handle the first term. Since $\sum_{i\in\mathcal{J}} w_i \leqslant 1$, we have in particular

$$\left|\left\{i\in\mathcal{J}\colon \varepsilon'\leqslant M w_i^{2+\frac{1}{p}}\right\}\right| \cdot \left(\frac{\varepsilon'}{M}\right)^{\frac{p}{2p+1}} \leqslant \sum_{i\in\mathcal{J},\varepsilon'\leqslant M w_i^{2+\frac{1}{p}}} w_i \leqslant 1 .$$

This shows that the number of intervals in the first term verifies

$$\left|\left\{i\in\mathcal{J}\colon \varepsilon'\leqslant M w_i^{2+\frac{1}{p}}\right\}\right| \leqslant M^{\frac{p}{2p+1}} (\varepsilon')^{-\frac{p}{2p+1}} , \tag{5.20}$$

which implies that the first term of (5.19) is bounded as

$$\sum_{i\in\mathcal{J},\varepsilon'\leqslant M w_i^{2+\frac{1}{p}}} (\varepsilon')^p \leqslant M^{\frac{p}{2p+1}} (\varepsilon')^{\frac{2p^2}{2p+1}} . \tag{5.21}$$

Now, let us bound from above the second term of the sum. Since $x\mapsto x^{2p+1}$ is strictly convex on $[0,1]$, Lemma 5.19 (in the appendix) applied with $\alpha = (\varepsilon'/M)^{p/(2p+1)}$ and $\beta \leqslant \sum_{i\in\mathcal{J}} w_i \leqslant 1$ states that the second term of (5.19) is bounded in the following way:

$$M^p \sum_{i\in\mathcal{J},\varepsilon'>M w_i^{2+\frac{1}{p}}} w_i^{2p+1} \leqslant 2M^p\alpha^{2p} = 2M^p\left(\frac{\varepsilon'}{M}\right)^{\frac{2p^2}{2p+1}} = 2M^{\frac{p}{2p+1}}(\varepsilon')^{\frac{2p^2}{2p+1}} . \tag{5.22}$$

Substituting the two upper bounds (5.21) and (5.22) into (5.19), we have

$$\phi((w_i)_{i\in\mathcal{J}}) \leqslant 3M^{\frac{p}{2p+1}}(\varepsilon')^{\frac{2p^2}{2p+1}} ,$$

which substituted into (5.18) yields

$$\begin{aligned} \|\widehat{f}_t - f\|_p^p &\leqslant \min\{K,t\}(\varepsilon')^p + 3M^{\frac{p}{2p+1}}(\varepsilon')^{\frac{2p^2}{2p+1}} \\ &\leqslant \min\{\varepsilon^{-\alpha},t\}c^p\varepsilon^{\gamma p} + 3M^{\frac{p}{2p+1}}c^{\frac{2p^2}{2p+1}}\varepsilon^{\frac{2\gamma p^2}{2p+1}} . \end{aligned} \tag{5.23}$$

Now, we finalize the proof by considering three cases depending on the value of α and by optimizing γ and c for each situation.

- *Case 1:* $0\leqslant\alpha\leqslant 1/2$. Then, $\varepsilon^{-\alpha}\leqslant\varepsilon^{-1/2}$, which substituted in (5.23), implies

$$\|\widehat{f}_t - f\|_p^p \leqslant c^p\varepsilon^{\gamma p-1/2} + 3M^{\frac{p}{2p+1}}c^{\frac{2p^2}{2p+1}}\varepsilon^{\frac{2\gamma p^2}{2p+1}} .$$

Choosing $\gamma = 1+\frac{1}{2p}$ yields

$$\|\widehat{f}_t - f\|_p^p \leqslant \left(c^p + 3M^{\frac{p}{2p+1}}c^{\frac{2p^2}{2p+1}}\right)\varepsilon^p .$$

The choice $c = \min\{2^{-1/p}, (6M)^{-\frac{1}{2p}}\}$ implies for $t \geqslant \tau_\varepsilon$

$$\|\widehat{f}_t - f\|_p \leqslant \varepsilon\,,$$

and by (5.14), the number of required evaluations is of order

$$\tau_\varepsilon = \widetilde{\mathcal{O}}\Big(c^{-p}\varepsilon^{-\frac{\gamma p}{p+1}}\Big) = \widetilde{\mathcal{O}}\Big(\varepsilon^{-1+\frac{1}{2p+2}}\Big)\,,$$

which concludes the first statement of the theorem.

- *Case 2:* $\alpha \geqslant 1$. Then, one may then use Theorem 5.3, which does note use the piecewise-regularity assumption of f, and implies that $\|\widehat{f}_t - f\|_p \leqslant \varepsilon$ for $t \geqslant \tau_\varepsilon$ with

$$\tau_\varepsilon = \mathcal{O}\big(\log(1/\varepsilon)^2 \mathcal{N}_p(f,\varepsilon)\big) = \mathcal{O}\big(\log(1/\varepsilon)^2 \varepsilon^{-1}\big) = \widetilde{\mathcal{O}}\Big(\varepsilon^{-1+\left(\frac{1-\alpha}{1+p}\right)_+}\Big)\,.$$

- *Case 3:* $1/2 \leqslant \alpha \leqslant 1$. Then, (5.23) yields

$$\|\widehat{f}_t - f\|_p^p \leqslant c^p \varepsilon^{\gamma p - \alpha} + 3M^{\frac{p}{2p+1}} c^{\frac{2p^2}{2p+1}} \varepsilon^{\frac{2\gamma p^2}{2p+1}}\,.$$

Substituting the choice $\gamma = 1 + \frac{\alpha}{p}$ into (5.23) further entails

$$\|\widehat{f}_t - f\|_p^p \leqslant c^p \varepsilon^p + 3M^{\frac{p}{2p+1}} c^{\frac{2p^2}{2p+1}} \varepsilon^{\left(\frac{2p+2\alpha}{2p+1}\right)p} \leqslant \Big(c^p + 3M^{\frac{p}{2p+1}} c^{\frac{2p^2}{2p+1}}\Big)\varepsilon^p\,.$$

Similary to the first case, choosing $c = \min\{2^{-1/p}, (6M)^{-\frac{1}{2p}}\}$ implies

$$\|\widehat{f}_t - f\|_p \leqslant \varepsilon\,,$$

and for any $t \geqslant \tau_\varepsilon$ of order

$$\tau_\varepsilon = \widetilde{\mathcal{O}}\Big(c^{-p}\varepsilon^{-\frac{\gamma p}{p+1}}\Big) = \widetilde{\mathcal{O}}\Big(\varepsilon^{-1+\frac{1-\alpha}{1+p}}\Big)\,.$$

This concludes the proof. □

5.B.6 Proof of Proposition 5.12

Proposition 5.12. Let $p \geqslant 1$, $\varepsilon \in (0,1)$ and $\alpha > 0$. Then, for any deterministic adaptive algorithm $\mathcal{A}$ and for any

$$t < (2\varepsilon)^{-1+\left(\frac{1-\alpha}{1+p}\right)_+} - 1\,,$$

there exists a non-decreasing piecewise-affine function $f : [0,1] \to [0,1]$ with at most $\max\{2, \lceil \varepsilon^{-\alpha} \rceil\}$ discontinuities, such that $\|f - \widehat{f}_t\|_p > \varepsilon$.

Proof. Let $p \geqslant 1$, $\varepsilon > 0$ and $\alpha > 0$. Let $\mathcal{A}$ be an adaptive algorithm and fix $t \geqslant 1$ a number of evaluations. We aim to design a function f with at most $K = \lceil \varepsilon^{-\alpha} \rceil$ discontinuities such that $\|f - \widehat{f}_t\|_p > \varepsilon$ if t is sufficiently small.

Let $g : x \to x$ be the identity function on $[0,1]$. Let $x_0 = 0$ and $x_{t+1} = 1$ and denote by $0 \leqslant x_1 \leqslant \ldots \leqslant x_t \leqslant 1$ the points that the algorithm would have chosen after t evaluations,

if it was applied on g and by $\widehat{f}_t$ its estimation. Then, we define two functions g_- and g_+ such that

$$\|g_- - g_+\|_p^p \geqslant \min\{Kt^{-1+p}, t^{-p}\}.$$

Let $K_- = \min\{K, t+1\}$. For $i \in \{1, \dots, t+1\}$, we define $w_i = x_i - x_{i-1}$ the width of the i-th interval. Let $\mathcal{J}$ denotes the set of indexes that correspond to the K_- largest intervals (i.e., such that $w_i \geqslant w_j$ for all $i \in \mathcal{J}$ and $j \notin \mathcal{J}$ and $|\mathcal{J}| = K_-$). Then, we define for all $x \in [0,1]$

$$g_-(x) = \begin{cases} x & \text{if } \exists i \notin \mathcal{J}, \text{ such that } x \in [x_{i-1}, x_i] \\ x_{i-1} & \text{if } x \in [x_{i-1}, x_i) \text{ for } i \in \mathcal{J} \end{cases}$$

and

$$g_+(x) = \begin{cases} x & \text{if } \exists i \notin \mathcal{J}, \text{ such that } x \in [x_{i-1}, x_i] \\ x_i & \text{if } x \in (x_{i-1}, x_i] \text{ for } i \in \mathcal{J} \end{cases}.$$

Then, g_- and g_+ have at most K discontinuities and are such that $g_-(x_i) = g_+(x_i) = g(x_i) = x_i$ for all $i \in \{1, \dots, t\}$. Thus, since $\mathcal{A}$ is deterministic, the function estimation and the points chosen by $\mathcal{A}$ on g_- and g_+ after t evaluations would also respectively be $\widehat{f}_t$ and $x_1, \dots, x_t$.

Furthermore, we have

$$\begin{aligned}
\|g_- - g_+\|_p^p &= \int_0^1 |g_-(x) - g_+(x)|^p \,\mathrm{d}x \\
&= \sum_{i \in \mathcal{J}} \int_{x_{i-1}}^{x_i} |g_-(x) - g_+(x)|^p \,\mathrm{d}x \\
&= \sum_{i \in \mathcal{J}} w_i^{p+1} \\
&\geqslant K_- \Big(\frac{1}{K_-} \sum_{i \in \mathcal{J}} w_i\Big)^{p+1} && \leftarrow \quad \text{by Jensen's Inequality} \\
&\geqslant K_- \Big(\frac{1}{t+1} \sum_{i=1}^{t+1} w_i\Big)^{p+1} && \leftarrow \quad \text{by Definition of } \mathcal{J} \\
&= K_- (t+1)^{-(p+1)} && \leftarrow \quad \text{because } \sum_{i=1}^{t+1} w_i = 1.
\end{aligned}$$

By triangular inequality, this yields

$$\begin{aligned}
\max_{f \in \{g_-, g_+\}} \|\widehat{f}_t - f\|_p &\geqslant \frac{1}{2}\Big(\|\widehat{f}_t - g_-\|_p + \|\widehat{f}_t - g_+\|_p\Big) \\
&\geqslant \frac{1}{2}\|g_- - g_+\|_p \\
&\geqslant \frac{1}{2} K_-^{\frac{1}{p}} (t+1)^{-\frac{p+1}{p}}.
\end{aligned}$$

Therefore, $\max_{f \in \{g_-, g_+\}} \|\widehat{f}_t - f\|_p > \varepsilon$ if

$$\frac{1}{2} K_-^{\frac{1}{p}} (t+1)^{-\frac{p+1}{p}} > \varepsilon$$

which, using $K_- \geqslant \min\{t+1, \varepsilon^{-\alpha}\}$, is satisfied for

$$t < \min\left\{(2\varepsilon)^{-1+(\frac{1-\alpha}{1+p})}, (2\varepsilon)^{-1}\right\} - 1\,.$$

Noting that g_- and g_+ have at most $K \leqslant \lceil \varepsilon^{-\alpha} \rceil$ discontinuities and that $g''(x) = 0$ elsewhere concludes the proof. □

5.B.7 Proof of Proposition 5.13

Proposition 5.13. Let $\varepsilon \in (0, 1/12)$. Then, there exists piecewise-C^2 function f_ε with one C^1-singularity, such that there exists $t \geqslant 2^{-7}\varepsilon^{-3/4}$ with $\left\|\widehat{f}_t - f_\varepsilon\right\|_1 > \varepsilon$ where $\widehat{f}_t$ is the GreedyBox approximation at t evaluations.

Proof. Let $k \in \mathbb{N}$ that will be chosen later by the analysis. Set $s = 2^{3k}$, $s' = 2^{2k}$ and $t = \frac{s+s'}{2}$.

Step 1. Design of a worst-case function We will design a worst-case function f^t that will cause GreedyBox to incur a large L^1-error after t iterations. The function will consist of two components: one for $x \leqslant 1/2$ that oscillates with a second derivative $|(f^t)''(x)| = 1$, and another that is linear for $x \geqslant 1/2$. f^t has a continuous derivative and its second derivative is piecewise-continuous wit $K = \frac{s'}{2}$ singularities. An illustration is given in Figure 5.2 for $k = 2$ ($t = 40$).

Let us first design the first oscillating part, that we call $g_{s'}$, and is defined recursively for all $x \in [0,1]$ by

$$g_{s'}(x) = \begin{cases} x^2 & \text{if } x \in \left[0, \frac{1}{s'}\right] \\ -\left(x - \frac{1}{s'}\right)^2 + \frac{2}{s'} \times \left(x - \frac{1}{s'}\right) + \left(\frac{1}{s'}\right)^2 & \text{if } x \in \left[\frac{1}{s'}, \frac{2}{s'}\right] \\ g_{s'}\left(x - \frac{2i}{s'}\right) + 2i\left(\frac{1}{s'}\right)^2 & \text{if } x \in \left[\frac{2i}{s'}, \frac{2(i+1)}{s'}\right], i \in \left\{1, \ldots, \frac{s'}{2} - 1\right\}. \end{cases} \tag{5.24}$$

Then, we define f^t by: for all $x \in [0,1]$

$$f^t(x) = \mathbb{1}_{\{x \leqslant 1/2\}} g_{s'}(x) + \mathbb{1}_{\{x > 1/2\}} \left(x - \frac{1}{2} + \frac{1}{2s'}\right). \tag{5.25}$$

Step 2. Expression of the approximation $\widehat{f}_t$ provided by GreedyBox after t iterations on f^t. Let us understand how GreedyBox behaves during the first t iterations when given this function. Once done, we will be able to retrieve the expression of the approximation $\widehat{f}_t$ of f^t made by GreedyBox.

Remark that for $i \in \{0, \ldots, s'/2 - 1\}$, (during the first oscillating part), the area of the box $B_i = \left[\frac{i}{s'}, \frac{i+1}{s'}\right] \times \left[f^t\left(\frac{i}{s'}\right), f^t\left(\frac{i+1}{s'}\right)\right]$ is

$$\mathcal{A}_p(B_i) = \frac{1}{s'} \times \frac{1}{s'^2} = \frac{1}{s'^3} = \frac{1}{s^2}.$$

Similarly, for $j \in \{\frac{s}{2}, \ldots, s-1\}$, (during the linear part), the area of the box $B_j = \left[\frac{j}{s}, \frac{j+1}{s}\right] \times \left[f^t\left(\frac{j}{s}\right), f^t\left(\frac{j+1}{s}\right)\right]$ is

$$\mathcal{A}_p(B_j) = \frac{1}{s^2}.$$

Thus, because GreedyBox tends to equalize the areas of the different boxes, and because it maintains areas of the form $[i/2^j,(i+1)/2^j]$ for some j in $\mathbb{N}^*$ and $i \in \{0,\dots,2^j-1\}$ (it can only splits intervals in 2), the box cover we just described is a potential output of GreedyBox after $t=(s'+s)/2$ epochs.

Let us formally prove that it is precisely the case. For any $0 \leqslant j' \leqslant 3k$ and $i \in \{1,\dots,2^{j'-1}-1\}$, the area of the box $[i2^{-j'},(i+1)2^{-j'}] \times [\widehat{f}_t(i2^{-j'}) + \widehat{f}_t((i+1)2^{-j'})]$ is $1/8^{j'}$. Likewise, for any $0 \leqslant j \leqslant 2k$ and $i \in \{2^{j-2},\dots 2^{j-1}-1\}$, the area of the box $[i2^{-j},(i+1)2^{-j}] \times [\widehat{f}_t(i2^{-j}) + \widehat{f}_t((i+1)2^{-j})]$ is $1/4^j$. Thus, if for some epoch $t' \leqslant t$, one box on the left of $1/2$ has a width greater than $3k$, its area will be greater than or equal to $8/s^2$, and if one box on the right of $1/2$ has a width greater than $2k$, its area will be greater than or equal to $4/s^2$. In both cases, the area is strictly greater than $1/s^2$, and GreedyBox would choose at Line 2 to split this box rather than any other box that already has area $1/s^2$.

Splitting one by one boxes with area greater than $1/s^2$, GreedyBox obtains at time t the t boxes with exact area $1/s^2$ described previously. $\widehat{f}_t$ is the linear interpolation between all the points $(x_i, f^t(x_i))$ where x_i is the extremity of one of the box.

Step 3. Error of GreedyBox on f^t after t iterations. Now that we exhibited the exact value of $\widehat{f}_t$, we are left with computing the total error made by GreedyBox on f^t after t steps. Since f^t is linear on the interval $[1/2,1]$, $\widehat{f}_t$ equals f^t on this segment. Then,

$$\begin{aligned}
\|f^t - \widehat{f}_t\|_1 &= \int_0^{\frac{1}{2}} \left|f^t(x) - \widehat{f}_t(x)\right| \mathrm{d}x = \sum_{i=1}^{s'/2-1} \int_{\frac{i}{s'}}^{\frac{i+1}{s'}} \left|f^t(x) - \widehat{f}_t(x)\right| \mathrm{d}x \\
&= \frac{s'}{2} \int_0^{\frac{1}{s'}} \left|f^t(x) - \widehat{f}_t(x)\right| \mathrm{d}x = \frac{s'}{2} \int_0^{\frac{1}{s'}} \left|x^2 - \frac{x}{s'}\right| \mathrm{d}x \\
&= \frac{s'}{2} \left(\frac{1}{2s'^3} - \frac{1}{3s'^3}\right) = \frac{1}{12(s')^2}\,.
\end{aligned}$$

Step 4. Choice of k. We are left with choosing k as large as possible such that the above error is at least ε. That is,

$$\|f^t - \widehat{f}_t\|_1 > \varepsilon$$

which can be rewritten as

$$\frac{2^{-4k}}{12} = \frac{1}{12s'^2} > \varepsilon \qquad \Leftrightarrow \qquad k < \frac{1}{4}\log_2\left(\frac{1}{12\varepsilon}\right).$$

Thus choosing $k = \lfloor \frac{1}{4}\log_2\left(\frac{1}{12\varepsilon}\right) \rfloor$, yields

$$t = \frac{2^{2k}+2^{3k}}{2} \geqslant 2^{3k-1} \geqslant (12\varepsilon)^{-\frac{3}{4}} 2^{-4} > 2^{-7}\varepsilon^{-\frac{3}{4}}\,.$$

Note that the designed function has a number of C^2-singularities

$$K = \frac{s'}{2} = 2^{2k-1} \leqslant \frac{1}{2}(12\varepsilon)^{-1/2} \leqslant \varepsilon^{-1/2}\,,$$

but only has one C^1-singularity at $x=1/2$. This concludes the proof. □

5.B.8 Technical lemmas for regular functions

In this section, we establish two technical lemmas that are used in our analysis of Greedy-Box for piecewise-C^2 functions. The first lemma, presented below, is a property of strictly convex functions.

Lemma 5.19. *Let $n \in \mathbb{N}^*$, $\beta \in \mathbb{R}^+$, $\alpha \in [\beta/n, \beta]$ and f be a strictly convex function on $[0,\beta]$. Then*

$$\max\left\{\sum_{i=1}^n f(x_i) \colon \forall i, x_i \leqslant \alpha, \sum_{i=1}^n x_i = \beta\right\} = mf(\alpha) + f(\beta - m\alpha) + (n-m+1)f(0)\,,$$

and the maximum is reached for $x_1^\star = \ldots = x_m^\star = \alpha$, $x_{m+1}^\star = \beta - m\alpha$, $x_{m+2}^\star = \ldots = x_n^\star = 0$, where $m = \lfloor \beta/\alpha \rfloor$.

Proof. Let $x^\star \in \mathbb{R}^n$ be as in the statement of the theorem, and let us show that it is optimal. Let x be a real-number. Since only the sum of the $f(x_i)$ matters, we can assume without loss of generality that the x_i's are sorted in decreasing order. Now, assume that $x_m < \alpha = x_m^\star$, and let us show that $\sum_{i=1}^n f(x_i)$ is not a maximum.

Because the sum of the x_i's still needs to be equal to β, either $x_{m+1} > x_{m+1}^\star = \beta - m\alpha$, or $x_{m+2} > x_{m+2}^\star = 0$. Assume first that $x_{m+2} > x_{m+2}^\star$. Furthermore, assume that $\min\{x_m^\star - x_m, x_{m+2} - x_{m+2}^\star\} = x_m^\star - x_m$. This means that x_m is closer to $x_m^\star$ than x_{m+2} is from $x_{m+2}^\star = 0$. Let $x = x_m^\star$, $\lambda = \frac{x_m^\star - x_{m+2}}{2x_m^\star - x_m - x_{m+2}} \in (0,1)$ and $y = x_{m+2} + x_m - x_m^\star \in [0, x_{m+2})$. Then,

$$\begin{aligned} f(x_m) + f(x_{m+2}) &= f(\lambda x + (1-\lambda)y) + f((1-\lambda)x + \lambda y) \\ &< \lambda f(x) + (1-\lambda)f(y) + (1-\lambda)f(x) + \lambda f(y) \\ &\leqslant f(x_m^\star) + f(y)\,. \end{aligned}$$

Then for $x_m' = x$, $x_{m+2}' = y$ and $x_i' = x_i$ for $i \neq \{m, m+2\}$, $\sum_{i=1}^n f(x_i') < \sum_{i=1}^n f(x_i)$, which shows that $(x_i)_{1\leqslant i\leqslant n}$ is suboptimal. We can do the same kind of construction when $x_{m+1} > x_{m+1}^\star$ or when $\min\{x_m^\star - x_m, x_{m+2} - x_{m+2}^\star\} = x_{m+2} - x_{m+2}^\star$. All these different cases show that $\sum_{i=1}^n f(x_i)$ is maximized only when the largest possible amount of the x_i's are at the extremity of the constraint set, that is when $x_i = 0$ or $x_i = \alpha$. This concludes the proof. □

The second technical Lemma below recalls a classical result and provides an upper bound on the L^p-error achieved by an affine approximation of a C^2 function. In particular, this lemma implies a sample complexity of order $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$ for the trapezoidal rule to provide an ε-approximation in L^p norm for C^2 functions.

Lemma 5.20. *Let $a < b$. Assume that $f : [a,b] \to [0,1]$ is C^1 and piecewise-C^2 and such that $|f^{(2)}(x)| \leqslant M$ for all x where it is defined. Then,*

$$\int_a^b \left|\widehat{f}(x) - f(x)\right|^p \mathrm{d}x \leqslant \left(\frac{3M}{2}\right)^p (b-a)^{2p+1}\,,$$

where $\widehat{f}$ is the affine approximation defined for all $x \in [a,b]$ by:

$$\widehat{f}(x) = f(a) + \big(f(b) - f(a)\big)\frac{x-a}{b-a}\,.$$

Proof. Since f is piecewise-C^2 with bounded second derivative, f' is absolutely continuous, we can thus apply the Taylor-Lagrange Theorem with the integral form of the remainder with $k=1$. We have for all $x,y\in[a,b]$

$$|f(y)-f(x)-f'(x)(y-x)| = \Big|\int_x^y f''(u)(y-u)\,\mathrm{d}u\Big| \leqslant \frac{M}{2}(y-x)^2\,.$$

We now control the L^p-error of $\widehat{f}$. The above inequality yields that for all $x\in[a,b]$ there exists $r(x)$ such that $|r(x)|\leqslant \frac{M}{2}(x-a)^2$ and

$$f(x) = f(a)+f'(a)(x-a)+r(x)\,.$$

Thus, applying it with $x=b$ entails

$$\frac{f(b)-f(a)}{b-a} = f'(a)+\frac{r(b)}{b-a}\,,$$

which in turns implies that

$$\begin{aligned}|\widehat{f}(x)-f(x)| &= |f(a)+(x-a)\frac{(f(b)-f(a))}{b-a}-f(x)|\\ &= \big|(x-a)\frac{r(b)}{b-a}+r(b)-r(x)\big|\\ &\leqslant \frac{M}{2}\big[(b-a)(x-a)+(b-a)^2+(x-a)^2\big]\,.\end{aligned}$$

Therefore, with the change of variable $x=a+(b-a)u$, we get

$$\begin{aligned}\int_a^b |\widehat{f}(x)-f(x)|^p\,\mathrm{d}x &\leqslant \Big(\frac{M}{2}\Big)^p\int_a^b\big[(b-a)(x-a)+(b-a)^2+(x-a)^2\big]^p\,\mathrm{d}x\\ &= \Big(\frac{M}{2}\Big)^p(b-a)^{2p+1}\int_0^1(1+u+u^2)^p\,\mathrm{d}u\\ &\leqslant \Big(\frac{3M}{2}\Big)^p(b-a)^{2p+1}\,,\end{aligned}$$

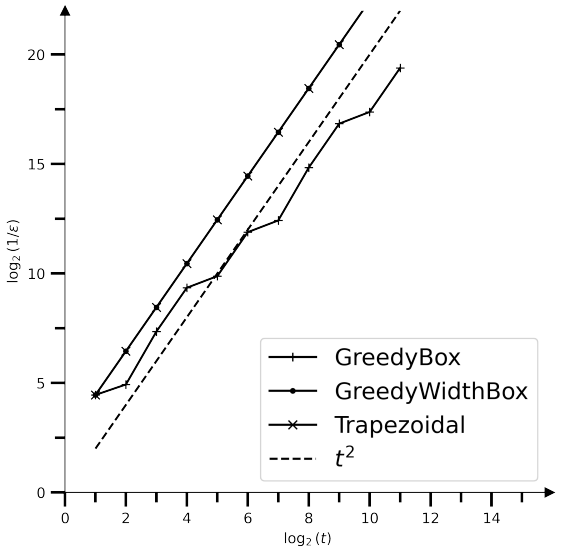
which concludes the proof. □

5.C Numerical experiments: the L^2-norm Case

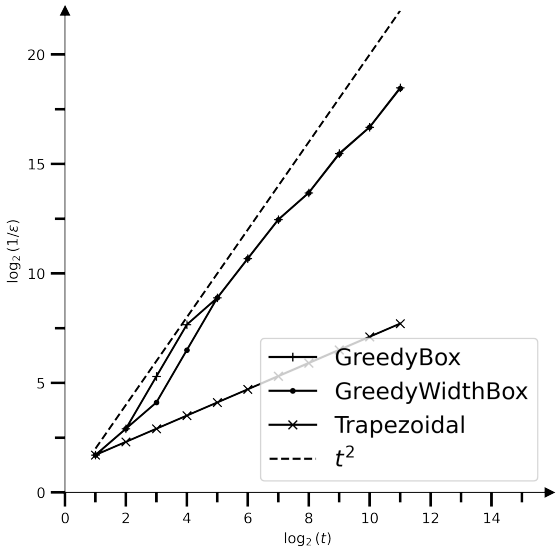
In the core of the paper, we gave some plots on the error of GreedyBox, and GreedyWidthBox as compared to the trapezoidal rule in Section 5.5. In this section, we complete the comparison by displaying some plots for the L^2-norm. In this case, the area of a box $B=[c^-,c^+]\times[y^-,y^+]$ is worth $(y^+-y^-)(c^+-c^-)^2$.

We can notice several interesting facts on the plots of Figure 5.6. First, the trapezoidal rule, and GreedyWidthBox behaves better than GreedyBox for the square function. This is the first case we could find where the trapezoidal rule has a better speed of convergence than GreedyBox. Another point is that on Figure 5.6b, the trapezoidal rule has a bound of order $t^{11/20}$. However, in Theorem 5.3, we proved that GreedyBox has an error at worst of the order of $\mathcal{N}(f,\varepsilon)$, which we proved to be smaller than $\lceil\varepsilon^{-1}\rceil$. Yet, for this example,

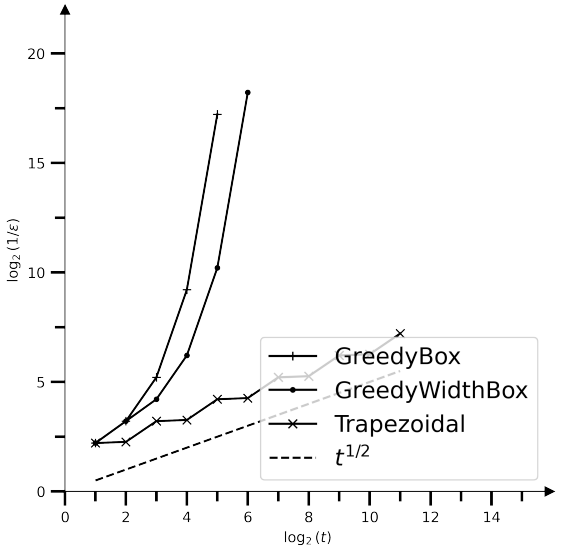
the trapezoidal does not converge in $\mathcal{O}(t)$, which proves that GreedyBox satisfies better worst case property than the trapezoidal rule for L^p-norm with $p > 1$. Figure 5.6c shows results similar to the L^1-norm. In, Figure 5.6d, we consider the L^2 for approximating $g^t : x \mapsto \frac{1}{2} f^{t^{9/10}}(2x)\mathbb{1}_{x \leqslant 1/2} + \mathbb{1}_{x > 1/2}$, where f^s is the function defined in equation (5.25) at $s = t^{9/10}$. The empirical rates seem to confirm once again the anticipated worst-case rates as determined by our analysis.



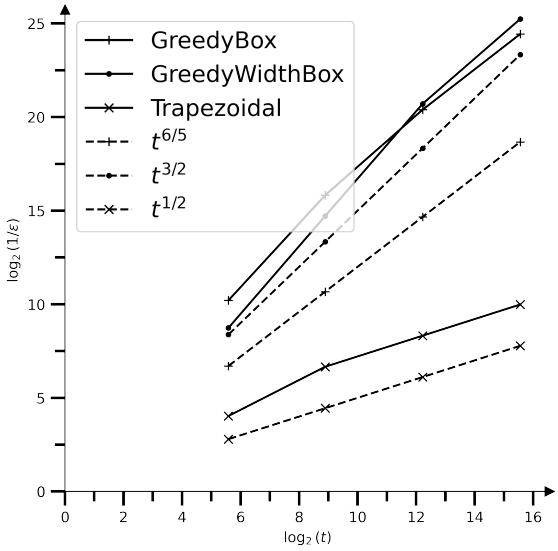
(a) Error rate in 2-norm on $f\colon x \mapsto x^2$



(b) Error rate in 2-norm on $f\colon x \mapsto x^{1/10}$



(c) Error rate in 2-norm on $f(x) = \mathbb{1}_{\{x \geqslant 0.3\}}$



(d) Error rate in 2-norm on g^t

Figure 5.6: Comparison of GreedyBox and GreedyWidthBox with the trapezoidal rule for the 2-norm.

Conclusion and Future Works

In this thesis, we explored diverse sequential algorithms, each with its set of challenges and applications.
In the first part, we studied OGDA, an algorithm that uses optimism in order to converge to a Nash equilibrium of a game. We provided a deep analysis of OGDA in the unconstrained bilinear case. We successfully answered the question of its convergence and speed in the zero-sum scenario, while providing preliminary results into its behavior for the general-sum case. Furthermore, we unveiled a nice use of general-sum games to solve in a faster way zero-sum games. However, a better understanding of OGDA and other optimistic algorithms remains to be dealt with when dealing with concave/convex functions. Another question of interest is whether this trick can also be used for concave/convex games (or to other larger class of games) and to any other algorithm, optimistic or not.

In the second part, we ventured into the study of the c.MF-DOO algorithm in the context of multi-fidelity zeroth-order optimization of Lipschitz functions with certificates. We also studied the behavior of a stochastic version of this algorithm, and gave an f-dependent upper bound for it. This naturally raises the question of whether a matching f-dependent lower bound can be proven. We can also wonder if the bounds we found can be generalized to the broader class of α-Hölder functions, or to functions with even weaker assumptions.

Finally, we studied GreedyBox, an algorithm for integral estimation and $L^p(\mu)$ approximation of monotone functions. We not only furnished f-dependent upper bounds of it, but also proved that it is optimal up to some logarithmic factors. We also showed that inherent to GreedyBox is a notion of certificate. It seems very likely that the current analysis can be extended to Lipschitz functions (considering trapezoidal boxes instead of rectangles). Another intriguing research direction is whether Greedybox can be extended to function with (known) bounded variations.

Bibliography

Jacob Abernethy, Kevin A Lai, Kfir Y Levy, and Jun-Kun Wang. Faster rates for convex-concave games. In *Conference On Learning Theory*, pages 1595–1625. PMLR, 2018.

Jacob D Abernethy, Kareem Amin, and Ruihao Zhu. Threshold bandits, with and without censored feedback. *Advances in Neural Information Processing Systems*, 29, 2016.

Nabila Aghanim, Yashar Akrami, Mark Ashdown, J Aumont, C Baccigalupi, M Ballardini, AJ Banday, RB Barreiro, N Bartolo, S Basak, et al. Planck 2018 results-VI. cosmological parameters. *Astronomy & Astrophysics*, 641:A6, 2020.

Natalia M Alexandrov, John E Dennis Jr, Robert Michael Lewis, and Virginia Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural optimization*, 15(1):16–23, 1998.

Ioannis Anagnostides, Gabriele Farina, Ioannis Panageas, and Tuomas Sandholm. Optimistic mirror descent either converges to Nash or to strong coarse correlated equilibria in bimatrix games. *arXiv preprint arXiv:2203.12074*, 2022.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Kenneth Joseph Arrow, Leonid Hurwicz, Hirofumi Uzawa, and Hollis Burnley Chenery. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

Jean-Yves Audibert, Sébastien Bubeck, et al. Minimax policies for adversarial and stochastic bandits. In *COLT*, volume 7, pages 1–122, 2009.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

François Bachoc, Tom Cesari, and Sébastien Gerchinovitz. Instance-dependent bounds for zeroth-order Lipschitz optimization with error certificates. *Advances in Neural Information Processing Systems*, 34:24180–24192, 2021.

Nikolai Sergeevich Bakhvalov. On the optimality of linear methods for operator approximation in convex classes of functions. *USSR Computational Mathematics and Mathematical Physics*, 11(4):244–249, 1971.

NS Bakhvalov. On approximate computation of integrals, vestnik mgu, ser. *Math. Mech. Astron. Phys. Chem*, 4(3):18, 1959.

Peter L Bartlett, Victor Gabillon, and Michal Valko. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Algorithmic Learning Theory*, pages 184–206. PMLR, 2019.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Gérard Biau, Benoît Cadre, Maxime Sangnier, Ugo Tanielian, et al. Some theoretical properties of gans. *Annals of Statistics*, 48(3):1539–1566, 2020.

Gérard Biau, Maxime Sangnier, and Ugo Tanielian. Some theoretical insights into Wasserstein gans. *Journal of Machine Learning Research*, 2021.

J. Frédéric Bonnans. *Convex and Stochastic Optimization*. Universitext. Springer Cham, 2019.

Clément Bouttier, Tommaso Cesari, Mélanie Ducoffe, and Sébastien Gerchinovitz. Regret analysis of the Piyavskii-Shubert algorithm for global Lipschitz optimization, 2020. Preprint available as arXiv:2002.02390.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Helmut Brass and Knut Petras. *Quadrature theory: the theory of numerical integration on a compact interval*. Number 178 in Mathematical Surveys and Monographs. American Mathematical Soc., 2011.

George W Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1):374, 1951.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015. ISSN 1935-8237.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12(5), 2011a.

Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu. Lipschitz bandits without the Lipschitz constant. In *Algorithmic Learning Theory*, pages 144–158, Berlin, Heidelberg, 2011b. Springer Berlin Heidelberg.

Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10):2879–2904, 2011.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

Prem Chandra. Trigonometric approximation of functions in Lp-norm. *Journal of Mathematical Analysis and Applications*, 275(1):13–26, 2002.

Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2013.

Yu.M. Danilin. Estimation of the efficiency of an absolute-minimum-finding algorithm. *USSR Computational Mathematics and Mathematical Physics*, 11(4):261–267, 1971. ISSN 0041-5553.

Prithviraj Dasgupta and Joseph Collins. A survey of game theoretic approaches for adversarial machine learning in cybersecurity tasks. *AI Magazine*, 40(2):31–43, 2019.

Constantinos Daskalakis and Qinxuan Pan. A counter-example to Karlin's strong conjecture for fictitious play. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2014.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *Innovations in Theoretical Computer Science*, 2019.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJJySbbAZ.

Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1466–1478, 2021.

Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1): 1281–1316, 2014.

Koen Dewettinck, Alex De Visscher, Lidewij Deroo, and André Huyghebaert. Modeling the steady-state thermodynamic operation point of top-spray fluidized bed processing. *Journal of Food Engineering*, 39(2):131–143, 1999.

Jelena Diakonikolas, Constantinos Daskalakis, and Michael I Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2746–2754. PMLR, 2021.

David Eby, RC Averill, William F Punch, and Erik D Goodman. Evaluation of injection island GA performance on flywheel design optimisation. In *Adaptive Computing in Design and Manufacture: The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation*, pages 121–136. Springer, 1998.

Tim Erven, Wouter M Koolen, Steven Rooij, and Peter Grünwald. Adaptive hedge. *Advances in Neural Information Processing Systems*, 24, 2011.

Billy Fang, Adityanand Guntuboyina, and Bodhisattva Sen. Multivariate extensions of isotonic regression and total variation denoising via entire monotonicity and Hardy-Krause variation. *The Annals of Statistics*, 49(2):769 – 792, 2021.

M Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T Haftka. Review of multi-fidelity models. *arXiv*, 2016. preprint arXiv:1609.07196.

Côme Fiegel, Victor Gabillon, and Michal Valko. Adaptive multi-fidelity optimization with fast learning rates. In *International Conference on Artificial Intelligence and Statistics*, pages 3493–3502. PMLR, 2020.

Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan 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*, SODA '05, page 385–394, USA, 2005. Society for Industrial and Applied Mathematics.

R Fletcher, JA Grant, and MD Hebden. The calculation of linear best Lp approximations. *The Computer Journal*, 14(3):276–279, 1971.

R Fletcher, JA Grant, and MD Hebden. Linear minimax approximation as the limit of best Lp-approximation. *SIAM Journal on Numerical Analysis*, 11(1):123–136, 1974.

Alexander IJ Forrester, Andy J Keane, and Neil W Bressloff. Design and analysis of "noisy" computer experiments. *AIAA journal*, 44(10):2331–2339, 2006.

Alexander IJ Forrester, András Sóbester, and Andy J Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: mathematical, physical and engineering sciences*, 463(2088):3251–3269, 2007.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and economic behavior*, 5(3):320–367, 1993.

Aurélien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376. JMLR Workshop and Conference Proceedings, 2011.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

Karl Geiselhart, Lori Ozoroski, James Fenbert, Elwood Shields, and Wu Li. Integration of multifidelity multidisciplinary computer codes for design and analysis of supersonic aircraft. In *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 465, 2011.

Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Gene H Golub and John H Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

S Graf and E Novak. The average error of quadrature formulas for functions of bounded variation. *The Rocky Mountain journal of mathematics*, 20(3):707–716, 1990.

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

Hédi Hadiji, Sarah Sachs, Tim van Erven, and Wouter M Koolen. Towards characterizing the first-order query complexity of learning (approximate) nash equilibria in zero-sum matrix games. *arXiv preprint arXiv:2304.12768*, 2023.

Pierre Hansen, Brigitte Jaumard, and Shi-Hui Lu. On the number of iterations of Piyavskii's global optimization algorithm. *Mathematics of Operations Research*, 16(2): 334–350, 1991.

Pierre Hansen, Brigitte Jaumard, and Shi-Hui Lu. Global optimization of univariate Lipschitz functions: I. survey and properties. *Mathematical Programming*, 55(1):251–272, 1992a.

Pierre Hansen, Brigitte Jaumard, and Shi-Hui Lu. Global optimization of univariate Lipschitz functions: II. new algorithms and computational comparison. *Mathematical Programming*, 55(1):273–292, 1992b.

Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(6), 2012.

José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

Aicke Hinrichs, Erich Novak, and Henryk Woźniakowski. The curse of dimensionality for the class of monotone functions and for the class of convex functions. *Journal of Approximation Theory*, 163(8):955–965, 2011.

Josef Hofbauer and William H Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

Matthias Horn. Optimal algorithms for global optimization in case of unknown Lipschitz constant. *Journal of Complexity*, 22(1):50–70, 2006.

Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: Convergence to spurious non-critical sets. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 4337–4348. PMLR, 2021.

Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. On the convergence of single-call stochastic extra-gradient methods. *Advances in Neural Information Processing Systems*, 32, 2019.

Deng Huang, Theodore T Allen, William I Notz, and R Allen Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382, 2006.

Adam Ibrahim, Waïss Azizian, Gauthier Gidel, and Ioannis Mitliagkas. Linear lower bounds and conditioning of differentiable games. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 4583–4593. PMLR, 13–18 Jul 2020.

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.

Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79: 157–181, 1993.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. *Advances in Neural Information Processing Systems*, 29, 2016a.

Kirthevasan Kandasamy, Gautam Dasarathy, Barnabás Póczos, and Jeff Schneider. The multi-fidelity multi-armed bandit. *Advances in Neural Information Processing Systems*, 29, 2016b.

Kirthevasan Kandasamy, Gautam Dasarathy, Junier Oliva, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Gaussian process bandit optimisation. *Journal of Artificial Intelligence Research*, 66:151–196, 2019.

Samuel Karlin. *Mathematical methods and theory in games, programming, and economics*, volume 2. Courier Corporation, 2003.

Carsten Katscher, Erich Novak, and Knut Petras. Quadrature formulas for multivariate convex functions. *Journal of Complexity*, 12(1):5–16, 1996.

J Kiefer. Optimum sequential search and approximation methods under minimum regularity assumptions. *Journal of the Society for Industrial and Applied Mathematics*, 5 (3):105–136, 1957.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4):1–77, 2019.

Kirill Kopotun and Alexei Shadrin. On k-monotone approximation by free knot splines. *SIAM journal on mathematical analysis*, 34(4):901–924, 2003.

Kirill A Kopotun, Dany Leviatan, and AV Prymak. Nearly monotone and nearly convex approximation by smooth splines in lp. *Journal of Approximation Theory*, 160(1-2): 103–112, 2009.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

David Krieg and Erich Novak. A universal algorithm for multivariate integration. *Foundations of Computational Mathematics*, 17:895–916, 2017.

Harold J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86:97–106, 1964.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, Cambridge, 2020.

Loic Le Gratiet. *Multi-fidelity Gaussian process regression for computer experiments*. PhD thesis, Université Paris-Diderot-Paris VII, 2013.

Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 907–915. PMLR, 2019.

Tianyi Lin, Chi Jin, and Michael I Jordan. Near-optimal algorithms for minimax optimization. In *Conference on Learning Theory*, pages 2738–2779. PMLR, 2020.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Andrea Locatelli and Alexandra Carpentier. Adaptivity to smoothness in X-armed bandits. In *Conference on Learning Theory*, pages 1463–1492. PMLR, 2018.

Aleksandar Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitri Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Cédric Malherbe and Nicolas Vayatis. Global optimization of Lipschitz functions. In *International Conference on Machine Learning*, pages 2314–2323. PMLR, 2017.

J. Močkus. On Bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk*, pages 400–404. Springer Berlin Heidelberg, 1975.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020a.

Aryan Mokhtari, Asuman E Ozdaglar, and Sarath Pattathil. Convergence rate of O(1/k) for optimistic gradient and extragradient methods in smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 30(4):3230–3251, 2020b.

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Advances in neural information processing systems*, 24, 2011.

Rémi Munos et al. From bandits to Monte-Carlo Tree Search: The optimistic principle applied to optimization and planning. *Foundations and Trends® in Machine Learning*, 7(1):1–129, 2014.

John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

Arkadij Semenovič Nemirovski and David Borisovich Yudin. Problem complexity and method efficiency in optimization. *Wiley*, 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Springer US, 2004.

Erich Novak. Quadrature formulas for monotone functions. *Proceedings of the American Mathematical Society*, 115(1):59–68, 1992.

Erich Novak. Quadrature formulas for convex classes of functions. In *Numerical Integration IV: Proceedings of the Conference at the Mathematical Research Institute, Oberwolfach, November 8–14, 1992*, pages 283–296. Springer, 1993.

Erich Novak. Optimal recovery and n-widths for convex classes of functions. *Journal of Approximation Theory*, 80(3):390–408, 1995.

Erich Novak. On the power of adaption. *Journal of Complexity*, 12(3):199–237, 1996.

Erich Novak and Ingo Roschmann. Numerical integration of peak functions. *Journal of Complexity*, 12(4):358–379, 1996.

Anargyros Papageorgiou. Integration of monotone functions of several variables. *Journal of Complexity*, 9(2):252–268, 1993.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3): 550–591, 2018.

Wei Peng, Yu-Hong Dai, Hui Zhang, and Lizhi Cheng. Training GANs with centripetal acceleration. *Optimization Methods and Software*, 35(5):955–973, 2020.

Alexander G Perevozchikov. The complexity of the computation of the global extremum in a class of multi-extremum problems. *USSR Computational Mathematics and Mathematical Physics*, 30(2):28–33, 1990.

Victor Picheny, David Ginsbourger, and Yann Richet. Noisy expected improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity. In *2nd International Conference on Engineering Optimization*, page 10 p., 2010.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and multidisciplinary optimization*, 48: 607–626, 2013.

SA Piyavskii. An algorithm for finding the absolute extremum of a function. *USSR Computational Mathematics and Mathematical Physics*, 12(4):57–67, 1972.

Leszek Plaskota and Grzegorz W Wasilkowski. Adaption allows efficient integration of functions with unknown singularities. *Numerische Mathematik*, 102:123–144, 2005.

Leszek Plaskota, Grzegorz Wasilkowski, and Yaxi Zhao. The power of adaption for approximating functions with singularities. *Mathematics of computation*, 77(264):2309–2338, 2008.

Leonid Denisovich Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28(5): 845–848, 1980.

Peter Z. G. Qian and C. F. Jeff Wu. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204, 2008.

Alexander Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 26, 2013.

Klaus Ritter, Grzegorz W Wasilkowski, and Henryk Woźniakowski. On multivariate integration for stochastic processes. In *Numerical Integration IV: Proceedings of the Conference at the Mathematical Research Institute, Oberwolfach, November 8–14, 1992*, pages 331–347. Springer, 1993.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 1952.

Julia Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

Rajat Sen, Kirthevasan Kandasamy, and Sanjay Shakkottai. Multi-fidelity black-box optimization with hierarchical partitions. In *International conference on machine learning*, pages 4538–4547. PMLR, 2018.

Rajat Sen, Kirthevasan Kandasamy, and Sanjay Shakkottai. Noisy blackbox optimization using multi-fidelity queries: A tree search approach. In *The 22nd international conference on artificial intelligence and statistics*, pages 2096–2105. PMLR, 2019.

Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. *Advances in neural information processing systems*, 19, 2006.

Xuedong Shang, Emilie Kaufmann, and Michal Valko. General parallel optimization without a metric. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98, pages 762–788. PMLR, 22–24 Mar 2019.

Bruno O Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 1958.

Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity Bayesian optimization with Gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3158–3167. PMLR, 2019.

James C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. Wiley-Interscience, 2003.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on information theory*, 58(5):3088–3095, 2012.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

Josef Stoer, Roland Bulirsch, R Bartels, Walter Gautschi, and Christoph Witzgall. *Introduction to numerical analysis*, volume 1993. Springer, 1980.

Aleksei G Sukharev. On the existence of optimal affine methods for approximating linear functionals. *Journal of Complexity*, 2(4):317–322, 1986.

Guangyong Sun, Guangyao Li, Shiwei Zhou, Wei Xu, Xujing Yang, and Qing Li. Multi-fidelity optimization for sheet metal forming process. *Structural and Multidisciplinary Optimization*, 44:111–124, 2011.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In *International Conference on Machine Learning*, pages 9334–9345. PMLR, 2020.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Michal Valko, Alexandra Carpentier, and Rémi Munos. Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*, pages 19–27. PMLR, 2013.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1998.

Jay M Ver Hoef and Noel Cressie. Multivariable spatial prediction. *Mathematical Geology*, 25:219–240, 1993.

Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, and Georgios Piliouras. Poincaré recurrence, cycles and spurious equilibria in gradient-descent-ascent for non-convex non-concave zero-sum games. *Advances in Neural Information Processing Systems*, 32, 2019.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische annalen*, 100(1): 295–320, 1928.

Vladimir Vovk. Aggregating strategies. In *Proceedings of 3rd Annu. Workshop on Comput. Learning Theory*, pages 371–383, 1990.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Junxiong Wang, Debabrota Basu, and Immanuel Trummer. Procrastinated tree search: Black-box optimization with delayed, noisy, and multi-fidelity feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10381–10390, 2022.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR, 2017.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations*, 2021.

Yue Wu. A randomized trapezoidal quadrature. *International Journal of Computer Mathematics*, 99(4):680–692, 2022.

Guojun Zhang and Yaoliang Yu. Convergence of gradient methods on bilinear zero-sum games. In *International Conference on Learning Representations*, 2020.

Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-based multi-fidelity Bayesian optimization. In *NIPS Workshop on Bayesian Optimization*, page 49. Journal of Machine Learning Research JMLR. org Cambridge, MA, 2017.

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