Online optimization and learning in games: Theory and applications

Panayotis Mertikopoulos

ONLINE OPTIMIZATION AND LEARNING IN GAMES: THEORY AND APPLICATIONS

PANAYOTIS MERTIKOPOULOS

Habilitation à Diriger des Recherches

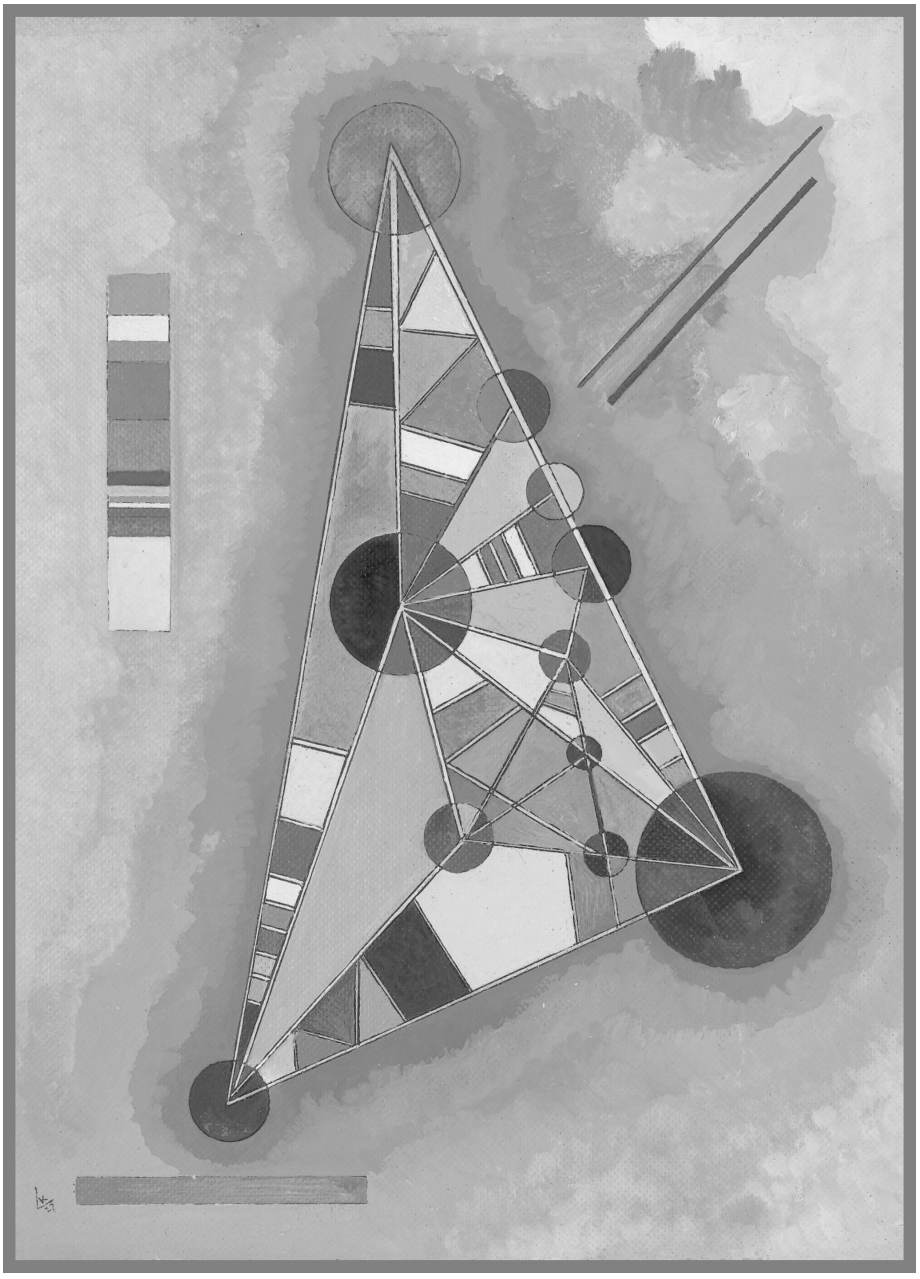
RAPPORTEURS

JÉRÔME BOLTE	Toulouse School of Economics & Université Toulouse 1 Capitole
NICOLÒ CESA-BIANCHI	Università degli Studi di Milano
SYLVAIN SORIN	Sorbonne Université

EXAMINATEURS

ERIC GAUSSIER	Université Grenoble Alpes
JOSEF HOFBAUER	Universität Wien
ANATOLI JUDITSKY	Université Grenoble Alpes
JÉRÔME RENAULT	Toulouse School of Economics & Université Toulouse 1 Capitole
NICOLAS VIEILLE	École des Hautes Études Commerciales de Paris (HEC Paris)

Université Grenoble Alpes
École doctorale MSTII
Spécialité: Informatique et Mathématiques Appliquées
Soutenue à Grenoble, le 20 décembre 2019



"*De plus j'aimais, et j'aime encore, les mathématiques pour elles-mêmes, comme n'admettant pas l'hypocrisie et le vague, mes deux bêtes d'aversion.*"

— Stendhal, *Vie de Henri Brulard*

PREFACE

The aim of this document is to give a bird's eye view of my research on multi-agent online learning. I do not claim to be comprehensive in this endeavor; rather, my guiding principle was to be comprehensible.

This had two main consequences: First, I had to leave out a fair number of research topics that I find equally exciting but did not otherwise fit the core of the present narrative. Second, I opted for a more conversational style, putting more emphasis on the results themselves rather than the technical trajectory that led to them. This might frustrate some readers who would seek to dive in the murky waters of the proofs and to analyze the technical contributions therein. I hope these readers will be satisfied by the list of references provided throughout this manuscript, and where all the relevant technical content can be found. Instead, my goal was to make the material presented herein accessible to non-experts in the field, to present a coherent narrative, and to explain what drives my research in these topics.

Chapter 2 is perhaps the clearest embodiment of this principle: it does not contain any original results per se, but rather intends to set the stage for the analysis to come. It reflects personal views – and biases – on the fundamentals of online learning and game theory, and I found it necessary to properly structure and position the rest of this document.

Chapters 3 and 4 comprise the theoretical backbone of my work and are aligned along two basic axes: continuous- vs. discrete-time considerations. From a practical viewpoint, the latter is often considered more interesting than the former: a system of stochastic differential equations can hardly be considered an implementable algorithm, and one could argue that modeling computer-aided decision processes as continuous-time dynamical systems is folly (on the surface at least). However, from a mathematical standpoint, the continuous- and discrete-time approaches comprise two synergistic research thrusts that dovetail in a unique and singular manner. Thanks to the theory of stochastic approximation (the glue that holds much of this manuscript together), insights gained in continuous time can be used to prove discrete-time results that would otherwise be inaccessible.

Chapters 5 and 6 focus on some applications of my work to high-performance computing (Chapter 5) and wireless communications (Chapter 6). I hesitated for a long time which applications to present and with what criteria to select them. In the end, I chose to focus on distributed computing and wireless networks because they were the closest in spirit to the material presented in the previous chapters, and because they provide an ideal playground for the theory developed therein. This meant that I had to leave out other equally interesting applications on generative adversarial networks and traffic routing, but this couldn't be helped.

Finally, Chapter 7 presents some perspectives and directions for future research that arise naturally from the body of work preceding it. If the style of the previous chapters can be characterized as conversational, this last chapter is one of vigorous hand-waving, aiming to find a light switch in the dark. The questions stated therein are of a fairly open

character, and I expect at least a few years to pass before any convincing answers are obtained.

Finally, Appendices A and B provide a series of biographical and bibliographical information. This is mostly intended to give some perspective of how the various ideas and questions evolved over time, and to provide some pointers to papers that treat a number of questions that could not be properly addressed within the rest of this manuscript.

DISCLAIMER. Before proceeding, I would feel remiss not to point out that this manuscript *is neither complete nor comprehensive* – nor does it purport to be. I have tried to provide pointers to the relevant literature throughout, but it is not possible to do an adequate (let alone comprehensive) survey of the state of the art for all the topics addressed herein. The interested reader should be fully aware of this and should treat this manuscript as an entry point to a much wider literature.

ACKNOWLEDGMENTS

The road leading to this document was long, full of turns and twists, and with its fair share of dead ends and disappointments. What I feel made this journey worthwhile was what I got from my friends and colleagues along the way.

First and foremost, I need to express my deep gratitude to Jérôme Bolte, Nicolò Cesa-Bianchi, and Sylvain Sorin, the *rapporteurs* of this HDR. They generously contributed an immense amount of time reviewing this manuscript in detail, and their keen and insightful remarks were invaluable. I am similarly thankful to my *examinateurs*, Eric Gaussier, Josef Hofbauer, Anatoli Juditsky, Jérôme Renault, and Nicolas Vieille: sacrificing their time and energy to make the trip to Grenoble in the middle of a nation-wide railway strike is, perhaps, the least reason for which I am grateful to all of them.

Of course, this document would never have existed without my extended academic family, their continued input, and their drive. I am particularly grateful to Rida for his constant bombardment of ideas over the years; to Veronica for her endless enthusiasm and vigor; to Bill for his unsurpassed mastery of game dynamics, matched only by his attention to detail; to Marco and Roberto for being role models of clarity and depth of thought; to Zhengyuan and Mathias for relentleslly pushing the boundary of the envelope; to all my co-authors for the countless lively exchanges and enriching discussions over the years; to my students and post-docs (Amélie, Olivier, Kimon, Yu-Guan,…) for all the things they taught me; and to my colleagues in Grenoble (Arnaud, Bary, Bruno, Franck, Jérôme, Patrick,…) for providing an ideal environment to work in.

Last but not least, I need to devote a special thanks to my parents, Kallia and Vlassis, my sister Victoria, and to my closest friends (Alex, Athena, Daniel, Lenja, Marios,…) for their endless support and encouragement. Most of all however, I want to thank my wife, Tonia, for generously carrying me through all these sleepless nights, for keeping me tethered to reality, and for lovingly reminding me that the darkest hour is just before the dawn. All this is for her.

FINANCIAL SUPPORT. My research was partially supported by the French National Research Agency (ANR) under grants ORACLESS (ANR-16-CE33-0004-01) and GAGA (ANR-16-13-JS01-0004-01), the Huawei Flagship Program ULTRON, and the EU COST Action CA16228 "European Network for Game Theory" (GAMENET). For a complete list of awarded grants, see Appendix A.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

AI	artificial intelligence
APT	asymptotic pseudotrajectory
BDA	bandit dual averaging
BR	best response
CCE	coarse correlated equilibrium
CE	correlated equilibrium
DA	dual averaging
DASGD	distributed asynchronous stochastic gradient descent
DGF	distance-generating function
DSC	diagonal strict concavity
EGD	entropic gradient descent
EGT	evolutionary game theory
ES	evolutionary stability
ESS	evolutionarily stable state
EW	exponential weights
EXP3	exploration and exploitation with exponential weights
FTL	follow-the-leader
FTLL	follow-the-linearized-leader
FTRL	follow-the-regularized-leader
GAN	generative adversarial network
GD	gradient descent
HPC	high-performance computing
HR	Hessian–Riemannian
IWF	iterative water-filling
KW	Kiefer–Wolfowitz
LGD	lazy gradient descent
LMD	lazy mirror descent
i.i.d.	independent and identically distributed
ICT	internally chain transitive
IS	importance sampling
LFP	leader-following policy
l.s.c.	lower semi-continuous
MAB	multi-armed bandit
MAC	multiple access channel
MD	mirror descent
MIMO	multiple-input and multiple-output

MUI	multi-user interference
MW	multiplicative weights
MXL	matrix exponential learning
NE	Nash equilibrium
NI	Nikaido–Isoda
OU	Orstein–Uhlenbeck
OGD	online gradient descent
OMD	online mirror descent
PI	principal investigator
PPAD	polynomial parity arguments on directed graphs
RDP	repeated decision problem
SDA	stochastic dual averaging
SDE	stochastic differential equation
SFO	stochastic first-order oracle
SGD	stochastic gradient descent
SP	saddle-point
SPSA	simultaneous perturbation stochastic approximation
SVM	suppport vector machine
SWF	simultaneous water-filling
VI	variational inequality
VLS	very large scale
VS	variational stability
WF	water-filling

RELEVANT PUBLICATIONS

This manuscript contains contributions by the author from the papers listed below. More details can be found at the beginning of each section; a complete list of publications is presented in Appendix B.

[1] E. Veronica Belmega, Panayotis Mertikopoulos, Romain Negrel, and Luca Sanguinetti. Online convex optimization and no-regret learning: Algorithms, guarantees and applications. `https://arxiv.org/abs/1804.04529`, 2018.

[2] Mario Bravo and Panayotis Mertikopoulos. On the robustness of learning in games with stochastically perturbed payoff observations. *Games and Economic Behavior*, 103, John Nash Memorial issue:41–66, May 2017.

[3] Mario Bravo, David S. Leslie, and Panayotis Mertikopoulos. Bandit learning in concave N-person games. In *NIPS '18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[4] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. Learning with bandit feedback in potential games. In *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[5] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. Hedging under uncertainty: Regret minimization meets exponentially fast convergence. In *SAGT '17: Proceedings of the 10th International Symposium on Algorithmic Game Theory*, 2017.

[6] Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. Penalty-regulated dynamics and robust learning procedures in games. *Mathematics of Operations Research*, 40(3):611–633, August 2015.

[7] Joon Kwon and Panayotis Mertikopoulos. A continuous-time approach to online optimization. *Journal of Dynamics and Games*, 4(2):125–148, April 2017.

[8] Rida Laraki and Panayotis Mertikopoulos. Higher order game dynamics. *Journal of Economic Theory*, 148(6):2666–2695, November 2013.

[9] Rida Laraki and Panayotis Mertikopoulos. Inertial game dynamics and applications to constrained optimization. *SIAM Journal on Control and Optimization*, 53(5):3141–3170, October 2015.

[10] Panayotis Mertikopoulos and Aris L. Moustakas. The emergence of rational behavior in the presence of stochastic perturbations. *The Annals of Applied Probability*, 20(4):1359–1388, July 2010.

[11] Panayotis Mertikopoulos and Aris L. Moustakas. Learning in an uncertain world: MIMO covariance matrix optimization with imperfect feedback. *IEEE Trans. Signal Process.*, 64(1): 5–18, January 2016.

[12] Panayotis Mertikopoulos and William H. Sandholm. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324, November 2016.

[13] Panayotis Mertikopoulos and William H. Sandholm. Riemannian game dynamics. *Journal of Economic Theory*, 177:315–364, September 2018.

[14] Panayotis Mertikopoulos and Mathias Staudigl. Convergence to Nash equilibrium in continuous games with noisy first-order feedback. In *CDC '17: Proceedings of the 56th IEEE Annual Conference on Decision and Control*, 2017.

[15] Panayotis Mertikopoulos and Mathias Staudigl. On the convergence of gradient-like flows with noisy gradient input. *SIAM Journal on Optimization*, 28(1):163–197, January 2018.

[16] Panayotis Mertikopoulos and Yannick Viossat. Imitation dynamics with payoff shocks. *International Journal of Game Theory*, 45(1-2):291–320, March 2016.

[17] Panayotis Mertikopoulos and Zhengyuan Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, January 2019.

[18] Panayotis Mertikopoulos, E. Veronica Belmega, and Aris L. Moustakas. Matrix exponential learning: Distributed optimization in MIMO systems. In *ISIT '12: Proceedings of the 2012 IEEE International Symposium on Information Theory*, pages 3028–3032, 2012.

[19] Panayotis Mertikopoulos, E. Veronica Belmega, Aris L. Moustakas, and Samson Lasaulce. Distributed learning policies for power allocation in multiple access channels. *IEEE J. Sel. Areas Commun.*, 30(1):96–106, January 2012.

[20] Panayotis Mertikopoulos, E. Veronica Belmega, Romain Negrel, and Luca Sanguinetti. Distributed stochastic optimization via matrix exponential learning. *IEEE Trans. Signal Process.*, 65(9):2277–2290, May 2017.

[21] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *SODA '18: Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[22] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *ICLR '19: Proceedings of the 2019 International Conference on Learning Representations*, 2019.

[23] Steven Perkins, Panayotis Mertikopoulos, and David S. Leslie. Mixed-strategy learning with continuous action sets. *IEEE Trans. Autom. Control*, 62(1):379–384, January 2017.

[24] Luigi Vigneri, Georgios Paschos, and Panayotis Mertikopoulos. Large-scale network utility maximization: Countering exponential growth with exponentiated gradients. In *INFOCOM '19: Proceedings of the 38th IEEE International Conference on Computer Communications*, 2019.

[25] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W. Glynn. Stochastic mirror descent for variationally coherent optimization problems. In *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[26] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Peter W. Glynn, and Yinyu Ye. Distributed stochastic optimization with large delays. Under review, 2018.

[27] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Peter W. Glynn, Yinyu Ye, Jia Li, and Fei-Fei Li. Distributed asynchronous optimization with unbounded delays: How slow can you go? In *ICML '18: Proceedings of the 35th International Conference on Machine Learning*, 2018.

[28] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W. Glynn. On the convergence of mirror descent beyond stochastic convex programming. *SIAM Journal on Optimization*, forthcoming.

1

INTRODUCTION

Depending on the context, the word "learning" might mean very different things: in network science and control, it could mean changing the way resources are allocated for a particular task over time; in deep learning and artificial intelligence, it could mean training a neural network to discriminate between different objects in an image, or to generate new images altogether; in statistics, it could mean inferring the mapping from inputs to outputs in a complicated process (such as the response of a protein to a targeted stimulus); etc. The aim of this manuscript is to provide an in-depth look into the design and analysis of online learning algorithms in different contexts, both theoretical and applied: to examine what is and what isn't possible, to analyze the interactions between different learning frameworks, and to provide concrete results that can be exploited in practice.

1.1 CONTEXT AND POSITIONING

One common denominator that emerges in this highly diverse landscape is that learning invariably involves an agent that seeks to progressively improve their performance on a specific task – i.e., to "*learn*". This agent – the "learner" – could be something as abstract as an algorithm (e.g., an artificial neural net), or something as mundane as a commuter going to work each day. Still, irrespective of the nature of the learner, learning is typically achieved via a *feedback loop* of the following general form:

1. The agent interfaces with their *environment* (a computer network, a dataset, etc.) by selecting an *action* (a resource allocation scheme, a weight configuration, etc.).
2. The agent receives some *feedback* based on the quality of the chosen action and the state of the environment (e.g., the number of users in the network, the available datapoints, etc.).

An added complication to the above is that, in many practical applications, the actions of the learner may also affect the state of the environment, so this feedback loop might go both ways. This is perhaps best illustrated by two examples:

Example 1.1 (Traffic routing)**.** Consider a set of computer users with a set of traffic demands to be routed over a network (such as the Internet). If a user chooses to route a significant amount of traffic through a part of the network employed by other users, this part might become congested and users might end up switching to different routes. In so doing however, other, previously uncongested links might now become congested, so the first user would have to adapt to the new reality. In this way, every user in the network must both (*i*) learn which routes of the network are more suitable for their traffic demands; and (*ii*) learn to adapt to the behavior of other users that are simultaneously vying for the same resources.

Example 1.2 (Generative adversarial networks)**.** A *generative adversarial network* (GAN) is an artificial intelligence algorithm used in unsupervised machine learning to generate samples from an unknown, target distribution (e.g., images with sufficiently many

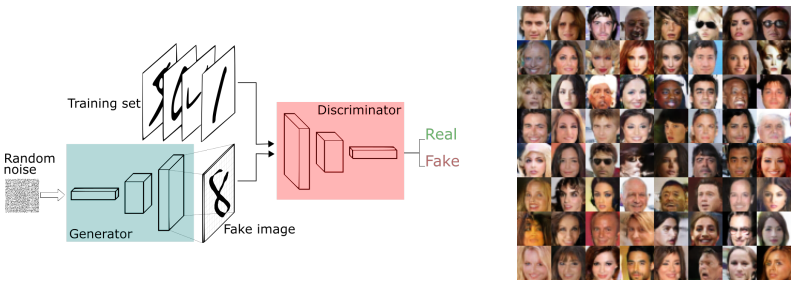


Figure 1.1: A typical GAN architecture and uncurated generated images taken from [102].

realistic characteristics to look authentic to human observers). Introduced in a seminal paper by Goodfellow et al. [56], a GAN consists of two neural networks competing against each other in a zero-sum game. One network – the *generator* – outputs candidate samples aiming to approximate the unknown target distribution, while the other network – the *discriminator* – evaluates the result based on a training set of instances taken from the true data distribution (for a schematic illustration, see Fig. 1.1). The objective of the generator is to fool the discriminator by providing samples that cannot be readily distinguished from the true distribution; at the same time, the discriminator seeks to adapt to the generator's evolution over time. In this way, each network plays simultaneously the role of the "learner" and of the "environment" (to the other network).

Both examples above can be construed as special cases of *multi-agent online learning:*[1] they comprise multiple interacting agents (or *players*), each with their individual actions, and seeking to attain possibly different objectives. As such, a fundamental question that arises in this context is the following:

Does learning lead to stability in multi-agent systems?

For instance, if all the users of a computer network follow a learning algorithm to try and learn the best route for their traffic demands, would that allow the system to converge to some "stable", steady state?

1.2 DIAGRAMMATIC OUTLINE

The rest of this manuscript aims to provide answers to this fundamental question in a range of different contexts, both practical and theoretical. For the reader's convenience, we provide a rough diagrammatic outline below and we rely on a series of margin notes and hyperlinks to facilitate the navigation of the manuscript.

Online optimization and game theory

CHAPTER 2. We begin in Chapter 2 by providing a gentle introduction to online optimization and game theory. The aim of this chapter is twofold: First, it represents an effort to make this manuscript as self-contained as possible by providing some fundamental results in the field. Second, we aim to establish a point of reference for the analysis to come by collecting all relevant definitions, prerequisites, and basic no-regret algorithms (such as online gradient descent, online mirror descent / dual averaging, etc.). The reader who is already familiar with the material presented here can safely skip ahead and use it only as a reference for notation and terminology.

Continuous-time analysis and results

CHAPTER 3. In this chapter, we flesh out a continuous-time skeleton for online optimization and learning in games. We discuss regret bounds in continuous time and how these overcome the corresponding minimax bounds for discrete time. We also

1 To maximize the number of applications treated in this manuscript, we do not revisit routing and GANs in the rest of this manuscript; for some of the author's work on these topics, see instead [102, 147].

introduce a continuous-time dynamical system induced in multi-player games by the no-regret learning algorithms of Chapter 2, and we discuss some basic properties of these dynamics – both negative and positive. Specifically, we discuss (*i*) the dynamics' Poincaré recurrence properties and their ramifications for convergence in zero-sum games (Section 3.2); (*ii*) the dynamics' long-run convergence and rationalizability properties, in both finite and continuous games (Section 3.3); and (*iii*) the robustness (and/or breakdown) of these properties in the presence of noise, modeled here as an Itô diffusion process (Section 3.4).

Discrete-time analysis and results

CHAPTER 4. In Chapter 4, we return to the discrete-time, no-regret framework of Chapter 2, and we examine which of the properties established in continuous time continue to hold in this bona fide algorithmic setting. More precisely, we discuss (*i*) the non-convergent behavior and the appearance of limit cycles under no-regret learning in zero-sum games in Section 4.1; (*ii*) the resolution of these phenomena in strictly monotone (or strictly coherent) games and games with dominated strategies or strict equilibria (Section 4.2); and (*iii*) the modifications to this analysis when the players of the game only have access to their in-game, realized payoffs (Section 4.3).

Applications to distributed optimization

CHAPTER 5. In this chapter, we examine a series of applications of the theory developed in the previous chapters to the solution of very large scale distributed optimization problems. We consider different multi-worker configurations of computer clusters (master-slave architectures or multi-core systems with shared memory), and we focus on the optimization properties of a distributed implementation of stochastic gradient descent in this setting, Our main point of interest is the algorithm's robustness to the delays incurred by different processors working at different speeds.

Applications to signal processing

CHAPTER 6. Continuing with a series of applications of the theory developed in Chapters 3 and 4, we discuss in this chapter a game-theoretic / distributed optimization framework for the problem of throughput maximization in multiple-input and multiple-output systems. The main contribution outlined in this chapter is the derivation and analysis of the matrix exponential learning algorithm, an efficient solution method for trace-constrained semidefinite optimization problems. This algorithm is heavily influenced by the game-theoretic analysis of Chapter 4 and is shown to provide state-of-the-art guarantees for multi-antenna systems and networks.

Finally, Chapter 7 contains some perspectives and direction for future research, while Appendices A and B provide a series of biographical and bibliographical information for the author. For the reader's convenience, we also provide a quick overview of the notational conventions used in this manuscript in the next section.

1.3 NOTATION AND TERMINOLOGY

Convex analysis

NOTATIONAL CONVENTIONS. Throughout what follows, $\mathcal{V}$ will denote a finite-dimensional real space with norm $\|\cdot\|$ and $\mathcal{X} \subseteq \mathcal{V}$ will be a closed convex subset thereof. Following standard conventions, we will write $\mathrm{ri}(\mathcal{X})$ for the relative interior of $\mathcal{X}$, $\mathrm{bd}(\mathcal{X})$ for its (relative) boundary, and $\mathrm{diam}(\mathcal{X}) = \sup\{\|x' - x\| : x, x' \in \mathcal{X}\}$ for its diameter. We will also write $\mathcal{Y} \equiv \mathcal{V}^*$ for the (algebraic) dual of $\mathcal{V}$, $\langle y, x \rangle$ for the canonical pairing between $y \in \mathcal{Y}$ and $x \in \mathcal{V}$, and $\|y\|_* \equiv \sup\{\langle y, x \rangle : \|x\| \leq 1\}$ for the dual norm of y in $\mathcal{Y}$.

Given an extended-real-valued function $f : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, its *effective domain* is defined as $\mathrm{dom}\, f = \{x \in \mathcal{V} : f(x) < \infty\}$ and its *subdifferential* at $x \in \mathrm{dom}\, f$ is given by $\partial f(x) = \{y \in \mathcal{V}^* : f(x') \geq f(x) + \langle y, x' - x \rangle \text{ for all } x' \in \mathcal{V}\}$. The doman of subdifferentiability of f is defined as $\mathrm{dom}\, \partial f \equiv \{x \in \mathcal{V} : \partial f(x) \neq \varnothing\}$. Finally, if $\partial f(x)$

is a singleton, we will say that f is *differentiable* at x and we will write $\nabla f(x)$ for the unique element thereof.

For all $x \in \mathcal{X}$, the *tangent cone* $\mathrm{TC}_{\mathcal{X}}(x)$ is defined as the closure of the set of all rays emanating from x and intersecting $\mathcal{X}$ in at least one other point. Dually to the above, the *polar cone* $\mathrm{PC}_{\mathcal{X}}(x)$ to $\mathcal{X}$ at x is defined as $\mathrm{PC}_{\mathcal{X}}(x) = \{y \in \mathcal{Y} : \langle y, z\rangle \leq 0 \text{ for all } z \in \mathrm{TC}_{\mathcal{X}}(x)\}$. For notational convenience, when $\mathcal{X}$ is understood from the context, we will drop it altogether and write more simply $\mathrm{TC}(x)$ and $\mathrm{PC}(x)$ instead.

Landau notation

In the sequel, we will also make heavy use of the Landau asymptotic notation $\mathcal{O}(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, etc. As a quick reminder, given two functions $f, g\colon \mathbb{R} \to \mathbb{R}$, we say that $f(t) = \mathcal{O}(g(t))$ if f grows no faster than g, i.e., there exists some positive constant $c > 0$ such that $|f(t)| < cg(t)$ for sufficiently large t (negative parts are ignored throughout). Conversely, we write $f(t) = \Omega(g(t))$ if f grows no slower than g, i.e., if $g(t) = \mathcal{O}(f(t))$. If we have both $f(t) = \mathcal{O}(g(t))$ and $f(t) = \Omega(g(t))$, we write $f(t) = \Theta(g(t))$; and if $\lim_{t\to\infty} f(t)/g(t) = 1$, we say that g grows as f and we write $f(t) \sim g(t)$ as $t \to \infty$. Finally, if $\limsup_{t\to\infty} f(t)/g(t) = 0$, we write $f(t) = o(g(t))$ and we say that f is *asymptotically dominated* by g.

Descent vs. ascent

A NOTE ON TERMINOLOGY. There is an unfortunate disconnect between game theory and optimization in terms of how objectives are formulated. In optimization, the objective is to *minimize* the incurred cost; in game theory, to *maximize* one's rewards. In turn, this creates a clash of terminology when referring to methods such as "gradient descent" or "mirror descent" in a maximization setting. To avoid going against the traditions of each field, we keep the usual conventions in place (minimization in optimization, maximization in game theory), and we rely on the reader to make the mental substitution of "descent" to "ascent" when needed.

Epicenes

Throughout this manuscript, we consider genderless agents and individuals. When an individual is to be singled out, we will consistently employ the pronoun "they" and its inflected or derivative forms. The debate between grammarians regarding the use of "they" as a singular epicene pronoun (with different editions of *The Chicago Manual of Style* famously providing different recommendations) is beyond the scope of this manuscript.

Part I

THEORY

2

PRELIMINARIES

Our aim in this introductory chapter is to discuss the basics of optimal decision-making in unknown environments – what are the criteria for optimality, the policies that attain them, etc. To do so, we take an approach based on two complementary viewpoints.

The first seeks to emulate the perspective of an agent that is faced with a recurring decision process but has no knowledge of its governing dynamics. We call this the "*unilateral viewpoint*" and we discuss it in detail in Sections 2.1 and 2.2. More precisely, Section 2.1 introduces the core framework of online optimization and the notion of regret which is central for our considerations; subsequently, in Section 2.2, we present an array of basic regret minimization algorithms and their performance guarantees.

The second viewpoint is more "holistic" and concerns several interacting agents whose decisions affect each other. The rules governing these interactions are still unknown to the agents, and the goal is to characterize those decisions that are simultaneously stable for each agent individually. We present this "*multi-agent viewpoint*" in Section 2.3: its main ingredients are non-cooperative games and the different solution concepts that arise in this context (Nash equilibrium, correlated equilibrium, etc.).

The natural bridging point between these two settings is the study of no-regret learning (a unilateral notion) in non-cooperative games (the quintessential element of the multi-agent viewpoint). This is the common unifying theme for most of this manuscript, and we examine it in detail in Chapters 3 and 4. The present chapter is meant to set the stage for the sequel by providing the core prerequisites for this analysis.

2.1 THE UNILATERAL VIEWPOINT: ONLINE OPTIMIZATION

 # This section incorporates material from the tutorial paper [13]

2.1.1 *The basic model*

Online optimization

Online optimization focuses on repeated decision problems (RDPs) where the objective is to minimize the aggregate loss incurred against a sequence of unknown loss functions. More precisely, the prototypical setting of online optimization can be summarized by the following sequence of events:

1. At every stage $t = 1, 2, \dots$, the optimizer selects an *action* X_t from a closed convex subset $\mathcal{X}$ of an ambient n-dimensional normed space $\mathcal{V}$.
2. Once an action has been selected, the optimizer incurs a loss $\ell_t(X_t)$ based on an (a priori) unknown loss function $\ell_t \colon \mathcal{X} \to \mathbb{R}$.
3. Based on the incurred loss and/or any other feedback received, the optimizer updates their action and the process repeats.

Based on the structural properties of ℓ_t, we have the following basic problem classes:

- *Online convex optimization:* each ℓ_t is assumed *convex*.

```
Require: action set 𝒳, sequence of loss functions ℓ_t: 𝒳 → ℝ
for t = 1, 2, ... do
   select X_t ∈ 𝒳                                  # action selection
   incur ℓ_t(X_t)                                  # incur loss
   update X_t ← X_{t+1}                            # update action
end for
```

Figure 2.1: Sequence of events in online optimization.

- *Online strongly convex optimization:* each ℓ_t is assumed *strongly convex*, i.e.,

$$\ell_t(x') \geq \ell_t(x) + \langle \nabla \ell_t(x), x' - x \rangle + \frac{\alpha_t}{2} \|x' - x\|^2 \tag{2.1}$$

  for some $\alpha_t > 0$ (called the *strong convexity modulus* of ℓ_t).

- *Online linear optimization:* each ℓ_t is assumed *linear*, i.e., of the form

$$\ell_t(x) = -\langle v_t, x \rangle \quad \text{for some } \textit{payoff vector } v_t \in \mathcal{V}^*. \tag{2.2}$$

Linear and strongly convex problems are both subclasses of the convex class, but they are otherwise disjoint; for convenience, we also assume throughout that each ℓ_t is differentiable[1] and that it attains its minimum in $\mathcal{X}$.

For concreteness, we discuss below some key examples of repeated decision problems (see also Fig. 2.1 for a pseudocode representation):

Static optimization

Example 2.1. Consider the static optimization problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \mathcal{X} \end{aligned} \tag{Opt}$$

where $f: \mathcal{X} \to \mathbb{R}$ is a static objective function. Viewed as an RDP, this corresponds to the case where the loss functions encountered by the optimizer are all equal to f, i.e.,

$$\ell_t(x) = f(x) \quad \text{for all } t = 1, 2, \dots \tag{2.3}$$

The optimality gap of a sequence of actions $X_t \in \mathcal{X}$ after T stages is then given by

$$\begin{aligned} \mathrm{Gap}(T) &= \sum_{t=1}^{T} f(X_t) - T \min_{x \in \mathcal{X}} f(x) = \sum_{t=1}^{T} \ell_t(X_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f(x) \\ &= \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]. \end{aligned} \tag{2.4}$$

This last quantity is known as the agent's *regret* and it plays a central role in the sequel.

Stochastic optimization

Example 2.2. Extending the above to problems involving randomness and uncertainty, consider the *stochastic* optimization problem

$$\begin{aligned} &\text{minimize} && f(x) \equiv \mathbb{E}[F(x; \omega)] \\ &\text{subject to} && x \in \mathcal{X} \end{aligned} \tag{Opt-S}$$

1 We adopt here the established convention of treating gradients as dual vectors. For book-keeping reasons, we will tacitly assume that ℓ_t is defined on an open neighborhood of $\mathcal{X}$ in $\mathcal{V}$; alternatively, in the convex case, if we view ℓ_t as an extended-real-valued function on $\mathcal{V}$ with effective domain $\operatorname{dom} \ell_t \equiv \{x \in \mathcal{V} : \ell_t(x) < \infty\} = \mathcal{X}$, we can simply assume that $\partial \ell_t$ admits a continuous selection, denoted by $\nabla \ell_t$. Either way, none of the results presented in the sequel depend on this device, so we do not make this assumption explicit.

where $F\colon \mathcal{X}\times\Omega \to \mathbb{R}$ is a stochastic objective defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In the RDP framework described above, it is assumed that an i.i.d. random sample $\omega_t \in \Omega$ is drawn at each stage $t = 1, 2, \dots$, and the loss function encountered by the optimizer is

$$\ell_t(x) = F(x; \omega_t) \quad \text{for all } t = 1, 2, \dots \tag{2.5}$$

As a result, the best that the optimizer could do on average would be to play a solution of (Opt-S); in turn, this leads to the mean optimality gap

$$\begin{aligned}\overline{\mathrm{Gap}}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(X_t)\right] - T \min_{x\in\mathcal{X}} f(x) = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(X_t)\right] - \min_{x\in\mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T} F(x;\omega_t)\right] \\ &\le \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(X_t)\right] - \mathbb{E}\left[\min_{x\in\mathcal{X}} \sum_{t=1}^{T} \ell_t(x)\right] \\ &= \mathbb{E}\left[\max_{x\in\mathcal{X}} \sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]\right]. \end{aligned} \tag{2.6}$$

Of course, if there is no randomness, this expression reduces to (2.4).

Example 2.3. As a third example, consider the saddle-point (SP) problem *Saddle-point problems*

$$\begin{aligned} &\text{minimize} && f(x) \equiv \max\nolimits_{\theta\in\Theta} \Phi(x;\theta) \\ &\text{subject to} && x \in \mathcal{X} \end{aligned} \tag{SP}$$

where θ is a parameter affecting the problem's value function $\Phi\colon \mathcal{X}\times\Theta \to \mathbb{R}$ in a manner beyond the optimizer's control. In other words, (SP) simply captures Wald's minimax optimization criterion of minimizing one's losses against the worst possible instance (i.e., attaining a certain security level no matter what).

In the associated RDP, $\theta_t \in \Theta$ is chosen at each stage $t = 1, 2, \dots$ by an abstract adversary, so the loss function encountered by the optimizer is

$$\ell_t(x) = \Phi(x; \theta_t) \quad \text{for all } t = 1, 2, \dots \tag{2.7}$$

Accordingly, the optimality gap relative to a minimax solution of (SP) is bounded as

$$\begin{aligned}\mathrm{Gap}(T) &= \sum_{t=1}^{T} \ell_t(X_t) - T \min_{x\in\mathcal{X}} f(x) = \sum_{t=1}^{T} \ell_t(X_t) - T \min_{x\in\mathcal{X}} \max_{\theta\in\Theta} \Phi(x;\theta) \\ &\le \sum_{t=1}^{T} \ell_t(X_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \Phi(x;\theta_t) = \sum_{t=1}^{T} \ell_t(X_t) - \min_{x\in\mathcal{X}} \sum_{t=1}^{T} \ell_t(x) \\ &= \max_{x\in\mathcal{X}} \sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]. \end{aligned} \tag{2.8}$$

Again, this expression is formally similar to the corresponding expression (2.4) for static optimization problems; we elaborate on this relation below.

2.1.2 *Regret and regret minimization*

In each of the above examples, there is a well-defined solution concept which could be viewed as a natural target point of the associated RDP (minimizers of f in Example 2.1, average minimizers in Example 2.2, and minimax solutions in Example 2.3). In general however, these concepts may not be meaningful because, unless more rigid assumptions are in place, there may be no fixed target point to attain, either static or in the mean.

This limitation is overcome by the notion of *regret*, which dates back at least to the work of Blackwell [19] and Hannan [57] in the 1950's:

Regret

Definition 2.1. The *regret* incurred by a sequence of actions $X_t \in \mathcal{X}$ against a sequence of loss functions $\ell_t \colon \mathcal{X} \to \mathbb{R}$, $t = 1, 2, \dots$, is defined as

$$\operatorname{Reg}(T) = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)] = \sum_{t=1}^{T} \ell_t(X_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x) \tag{2.9}$$

i.e., as the difference between the aggregate loss incurred by the agent after T stages and that of the best action in hindsight.

In words, the agent's regret contrasts the performance of the agent's policy X_t to that of an action $x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} \ell_t(x)$ which minimizes the total incurred loss over the horizon of play. On that account, the main goal in online optimization is to design causal, online policies that achieve *no regret*, i.e.,

No regret

$$\operatorname{Reg}(T) = o(T) \quad \text{for any sequence of loss functions } \ell_t,\ t = 1, 2, \dots \tag{2.10}$$

The performance of such a policy is then evaluated in terms of the actual *regret minimization* rate achieved, i.e., the precise expression in the $o(T)$ term above.

The situation becomes more complicated if the policy X_t is itself random – and, in particular, if its randomness is correlated to any randomness that might underlie ℓ_t. In that case, the expected regret of a policy X_t is defined as

Expected regret and pseudo-regret

$$\mathbb{E}[\operatorname{Reg}(T)] = \mathbb{E}\left[\max_{x \in \mathcal{X}} \sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]\right]. \tag{2.11}$$

This expression involves the expectation of a minimum which, in general, is difficult to compute. Instead, a more useful proxy for the regret in stochastic environments is the so-called *pseudo-regret*, defined here as

$$\overline{\operatorname{Reg}}(T) = \max_{x \in \mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]\right] \tag{2.12}$$

Since the maximum of the expectation of a family of random variables is majorized by the expectation of the maximum, we have

$$\overline{\operatorname{Reg}}(T) \le \mathbb{E}[\operatorname{Reg}(T)], \tag{2.13}$$

so the pseudo-regret is tighter as a worst-case guarantee.

In a stochastic setting, it is more natural to target the optimal action in expectation rather than the action which is optimal against the sequence of realized losses. Moreover, by standard Chernoff–Hoeffding arguments, the typical difference between the expected regret and the pseudo-regret is of the order of $\Theta(\sqrt{T})$. Thus, in general, one cannot hope to achieve an expected regret minimization rate better than $\mathcal{O}(\sqrt{T})$; by contrast, in several cases of interest, it is possible to attain much more refined bounds for the pseudo-regret. Because of this, $\overline{\operatorname{Reg}}(T)$ will be our principal figure of merit for regret minimization in the presence of randomness and/or uncertainty.

We close this section by revisiting some of the previous examples:

Value convergence

Example 2.4. Going back to the static framework of Example 2.1, Eq. (2.4) yields

$$\frac{1}{T} \sum_{t=1}^{T} f(X_t) = \min f + \frac{1}{T} \operatorname{Reg}(T). \tag{2.14}$$

Hence, if X_t is a no-regret policy, the sequence of function values $f(X_t)$ converges to $\min f$ in the Cesàro sense. In particular, if f is convex, Jensen's inequality shows that the time-averaged sequence

$$\bar{X}_T = \frac{1}{T}\sum_{t=1}^{T} X_t \tag{2.15}$$

achieves the value convergence rate

$$f(\bar{X}_T) \le \min f + \frac{1}{T}\operatorname{Reg}(T). \tag{2.16}$$

Likewise, in the stochastic setting of Example 2.2, we readily obtain

$$\mathbb{E}[f(\bar{X}_T)] \le \min f + \frac{1}{T}\overline{\operatorname{Reg}}(T) \tag{2.17}$$

provided that ω_t is independent of X_t.[2]

This mode of convergence is often referred to as "ergodic convergence" or Polyak–Ruppert averaging [117] and its study dates back at least to Novikoff [114]. In particular, if $\bar{X}_T$ is viewed as the output of an optimization algorithm generating the sequence of states $X_t \in \mathcal{X}$, the induced (pseudo-)regret directly reflects the algorithm's value convergence rate. We will revisit this issue several times in the sequel.

Multi-armed bandits

Example 2.5. Consider the following discrete variant of the stochastic optimization setting of Example 2.2. At each stage $t = 1, 2, \dots$, the optimizer selects an action a_t from some finite set $\mathcal{A} = \{1, \dots, n\}$. The reward of each arm $a \in \mathcal{A}$ is assumed to be an i.i.d. random variable $v_{a,t} \in [0,1]$ drawn from an unknown distribution P_a, and the aim is to choose the action with the highest mean reward as often as possible. Following Robbins [119], this is called a (stochastic) *multi-armed bandit* (MAB) problem in reference to the "arms" of a slot machine in a casino – a "bandit" in the colorful slang of the 1950's.

To quantify the above, let μ_a denote the mean value of the reward distribution P_a of the a-th arm, and let

$$\mu^* \equiv \max_{a\in\mathcal{A}} \mu_a \quad \text{and} \quad a^* \equiv \arg\max_{a\in\mathcal{A}} \mu_a \tag{2.18}$$

respectively denote the bandit's maximal mean reward and the arm that achieves it (a priori, there could be several such arms but, for simplicity, we assume here that there is only one). Then, if the agent selects $a \in \mathcal{A}$ at time t with probability $X_{a,t}$, the induced (pseudo-)regret after T stages will be

$$\begin{aligned}\overline{\operatorname{Reg}}(T) &= \mu^* T - \sum_{t=1}^{T} \mathbb{E}[v_{a_t,t}] = \max_{a\in\mathcal{A}} \sum_{t=1}^{T} \mathbb{E}[v_{a,t} - v_{a_t,t}] \\ &= \max_{x\in\Delta(\mathcal{A})} \mathbb{E}\left[\sum_{t=1}^{T} \langle v_t, x - X_t\rangle\right] = \max_{x\in\Delta(\mathcal{A})} \mathbb{E}\left[\sum_{t=1}^{T} [\ell_t(X_t) - \ell_t(x)]\right]\end{aligned} \tag{2.19}$$

where: (*i*) a_t denotes the arm played at time t; (*ii*) $v_t = (v_{a,t})_{a\in\mathcal{A}} \in \mathbb{R}^n$ is the *reward vector* of stage t (typically it is assumed that $-1 \le v_{a,t} \le 1$ for all t); and (*iii*) the loss functions ℓ_t are defined as $\ell_t(x) = -\langle v_t, x\rangle$.[3] Under this light, multi-armed bandits can be seen as (linear) stochastic optimization problems over the simplex $\mathcal{X} \equiv \Delta(\mathcal{A})$ of probability distributions over $\mathcal{A}$.

2 Recall here that ω_t is drawn after X_t, so this independence is not restrictive: for instance, this is always the case if ω_t is i.i.d. and X_t is predictable relative to the history $\sigma(\omega_1, \dots, \omega_{t-1})$ of ω_t up to stage $t-1$.

3 Since each ℓ_t is linear in x, maximizing over $\mathcal{A}$ or $\Delta(\mathcal{A})$ gives the same result.

2.2 NO-REGRET ALGORITHMS

 # This section incorporates material from the tutorial paper [13]

We now turn to the fundamental question underlying the online optimization framework discussed above:

Is it possible to achieve no regret? And, if so, at what rate?

Of course, any answer to these questions must depend on the specifics of the problem at hand – the assumptions governing the problem's loss functions, the information available to the optimizer, etc. In the rest of this section, we focus on obtaining simple answers in a range of different settings that arise in practice.

2.2.1 *Feedback assumptions*

Oracle feedback

We begin by specifying the type and amount of information available to the optimizer. Our starting point will be the so-called "oracle model" in which the optimizer gains access to each loss function via a black-box feedback mechanism (i.e., in a model-agnostic manner). Formally, given a (complete) probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and a measurable space of *signals* $\mathcal{S}$, an *oracle* for a function $f\colon \mathcal{X} \to \mathbb{R}$ is simply a map $\mathrm{Or}_f\colon \mathcal{X} \times \Omega \to \mathcal{S}$ that outputs a (random) signal $\mathrm{Or}_f(x;\omega) \in \mathcal{S}$ when called at $x \in \mathcal{X}$. Some important examples of such oracles are discussed below:

Full information

1. *Full information:* In this case, the space of signals $\mathcal{S}$ is a space of functions and $\mathrm{Or}_f(x) = f$. In other words, when called at a point $x \in \mathcal{X}$, a full-information oracle returns the *entire* function $f\colon \mathcal{X} \to \mathbb{R}$ (hence the name). [This oracle is assumed deterministic, so we suppress the argument ω.]

2. *Perfect n-th order information:* Oracles in this class return information on the n-th order derivates of the function at the input point $x \in \mathcal{X}$. In the "perfect information" case, when called at $x \in \mathcal{X}$, the oracle returns the tensor

$$\mathrm{Or}_f(x) = D^n f(x) = \left(\frac{\partial^n f}{\partial x_{i_1} \dots \partial x_{i_n}} \right)_{i_1,\dots,i_n=1,\dots,n} \tag{2.20}$$

   [Again, these oracles are assumed deterministic so the argument ω is suppressed.] By far the most widely used oracles of this type are the cases $n = 0, 1$ and 2:

   Bandit feedback
   - The case $n = 0$ gives $\mathrm{Or}_f(x) = f(x)$ and is known in the literature as *bandit feedback* (in reference to the multi-armed bandit problem of Example 2.5). It is most common in problems where scarcity of information plays a major role (such as adversarial and game-theoretic learning).

   Gradient feedback
   - The case $n = 1$ corresponds to perfect *gradient feedback*, i.e., $\mathrm{Or}_f(x) = \nabla f(x)$. This is most common in medium-to-small-scale optimization problems where exact gradient calculations are still within reach.

   Hessian feedback
   - The case $n = 2$ amounts to accessing to the Hessian matrix $\mathrm{Hess}(f(x))$ of f at x. Hessian calculations are very intensive in terms of computational power, so such oracles are most commonly encountered in relatively small-scale optimization problems that need to be solved to a high degree of accuracy.

3. *Noisy n-th order information:* Here, the setup is as before with the difference that the output of the oracle is random. Specializing directly to the cases that are most common in practice, we have:

- The case $n = 0$ corresponds to noisy function evaluations of the form $\mathrm{Or}_f(x;\omega) = f(x) + \xi(x;\omega)$ for some additive noise variable $\xi \in \mathbb{R}$. Oracles of this type are the norm in bandit convex optimization problems and problems where even the calculation of the objective function is computationally intensive. *Noisy bandit feedback*
- The case $n = 1$ provides noisy gradient evaluations of the form $\mathrm{Or}_f(x;\omega) = \nabla f(x) + U(x;\omega)$ for some observational noise variable $U \in \mathcal{V}^*$. This type of feedback is extremely common in distributed optimization problems with objectives of the form $f(x) = \sum_{i=1}^N f_i(x)$: taking a random sample $i \in \{1,\dots,N\}$ (or a minibatch thereof) and calculating its gradient gives a stochastic first-order oracle that is widely used in machine learning, signal processing, and data science. *Noisy gradient feedback*

Memory requirements

The informational content of an oracle can be gauged by the dimension of the signal space $\mathcal{S}$, which in turn provides an estimate of the memory required to store and process the received signal. In the full information case, $\mathcal{S}$ is typically infinite-dimensional, so such oracles are of limited practical interest (but are still very useful from a theoretical standpoint). At the other end of the spectrum, zeroth-order oracles have $\mathcal{S} = \mathbb{R}$, so they are the lightest in terms of memory requirements (but are otherwise the hardest to work with). Between these two extremes, n-th order oracles have $\mathcal{S} = (\mathcal{V}^*)^{\otimes n} \cong \mathbb{R}^{n^n}$, i.e., their memory requirements grow exponentially in n. This "curse of dimensionality" is one of the main reasons for the sweeping popularity of first-order methods in problems where the dimension of the ambient state space $\mathcal{V}$ becomes prohibitively large.

Stochastic first-order oracle feedback

In view of the above, a large part of our analysis will focus on *stochastic first-order oracles* (SFOs) that return possibly imperfect gradient measurements at the point where they are called. Specifically, we will assume that such an oracle is called repeatedly at a (possibly random) sequence of points $X_t \in \mathcal{X}$ and, at each stage $t = 1, 2, \dots$, returns a vector signal of the form

$$\nabla_t = \nabla \ell_t(X_t) + Z_t \tag{2.21}$$

where the "observational error" term Z_t captures all sources of uncertainty in the oracle.

Random vs. systematic errors

In more detail, to differentiate between "random" (zero-mean) and "systematic" (non-zero-mean) errors in ∇_t, it will be convenient to decompose Z_t as

$$Z_t = U_t + b_t \tag{2.22}$$

where U_t is zero-mean and b_t captures the mean value of Z_t. To define these two processes formally, we will subsume any randomness in ∇_t and ℓ_t in a joint event ω_t drawn from some (complete) probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Since this randomness is generated *after* the optimizer selects an action $X_t \in \mathcal{X}$ (cf. the sequence of events in Fig. 2.1), the processes ∇_t and Z_t are, a priori, not adapted to the history of X_t. More explicitly, writing $\mathcal{F}_t = \sigma(X_1, \dots, X_t)$ for the natural filtration of X_t, the processes b_t and U_t are defined as

$$b_t = \mathbb{E}[Z_t \,|\, \mathcal{F}_t] \qquad \text{and} \qquad U_t = Z_t - b_t \tag{2.23}$$

so, by definition, $\mathbb{E}[U_t \,|\, \mathcal{F}_t] = 0$.

SFO statistics

In view of all this, SFOs can be classified according to the following statistics:

1. *Bias:*
$$B_t = \mathbb{E}[\|b_t\|_*] \tag{2.24a}$$
2. *Variance:*
$$\sigma_t^2 = \mathbb{E}[\|U_t\|_*^2] \tag{2.24b}$$

Algorithm 2.1: Follow the regularized leader

Require: strongly convex regularizer $h\colon \mathcal{X} \to \mathbb{R}$; weight parameter $\gamma > 0$
for $t = 1, 2, \dots$ **do**
  play $X \leftarrow \arg\min_{x\in\mathcal{X}} \left\{\sum_{s=1}^{t} \ell_s(x) + \gamma^{-1} h(x)\right\}$ # choose action
  observe ℓ_t # get feedback
end for

3. *Second moment:*

$$M_t^2 = \mathbb{E}[\|\nabla_t\|_*^2] \tag{2.24c}$$

An oracle with $B_t = 0$ for all t will be called *unbiased*, and an oracle with $\lim_{t\to 0} B_t = 0$ will be called *asymptotically unbiased*; finally, an unbiased oracle with $\sigma_t = 0$ for all t will be called *perfect*. We will examine all these cases in detail in the sequel.

2.2.2 *Leader-following policies*

Follow the leader

The first no-regret candidate process that we will examine is based on the following simple principle: at time $t+1$, the optimizer plays the action that is optimal in hindsight up to (and including) stage t. This policy is known as *follow-the-leader* (FTL) and it can be formally described via the update rule:

$$X_{t+1} \in \arg\min_{x\in\mathcal{X}} \sum_{s=1}^{t} \ell_s(x) \tag{FTL}$$

with the usual convention $\sum_{t\in\varnothing} a_t = 0$ for the empty sum (i.e., X_1 is initialized arbitrarily).

Cover's impossibility principle

In terms of computational overhead, this policy requires a full information oracle (i.e., the knowledge of ℓ_t once X_t is chosen) and the ability to compute the arg min in the (FTL) update rule. Both requirements are significantly lighter in online linear optimization problems where each objective is of the form (2.2). However, even in this restricted setting, (FTL) incurs positive regret: a well-known example over $\mathcal{X} = [-1, 1]$ is provided by the sequence of alternating losses

$$\ell_t(x) = \begin{cases} -x/2 & \text{for } t = 1, \\ x & \text{if } t > 1 \text{ is even}, \\ -x & \text{if } t > 1 \text{ is odd}. \end{cases} \tag{2.25}$$

Against this sequence, (FTL) gives $X_t = \arg\min_{x\in[-1,1]} (-1)^{t-1} x/2 = (-1)^t$ for all $t > 1$, so the total incurred loss after T stages is $\sum_{t=1}^{T} \ell_t(X_t) = T - X_1/2 - 1$. By contrast, the constant policy $X_t = 0$ incurs zero loss for all t, implying in turn that $\mathrm{Reg}(T) \sim T$.

Follow the regularized leader

The main reason behind this failure is that (FTL) is too "aggressive". Indeed, if the cumulative loss function $\sum_{s=1}^{t} \ell_s$ exhibits significant oscillations between one stage and the next, (FTL) will also jiggle between extremes, and this behavior can be exploited by the adversary. One way to overcome this behavior is to introduce a penalty term which makes these oscillations less extreme (and hence, less exploitable); this leads to a policy known as *follow-the-regularized-leader* (FTRL) [135–137], and which can be stated as follows:

$$X_{t+1} = \arg\min_{x\in\mathcal{X}} \left\{ \sum_{s=1}^{t} \ell_s(x) + \frac{1}{\gamma} h(x) \right\}. \tag{FTRL}$$

[For a pseudocode implementation, see Algorithm 2.1.]

In the above, $h\colon \mathcal{X} \to \mathbb{R}$ is a *regularization* (or *penalty*) *function* and $\gamma > 0$ is a tunable parameter that adjusts the weight of the regularization term. By this token, to ensure

that this mechanism dampens oscillations, it is common to assume that the regularizer $h\colon \mathcal{X} \to \mathbb{R}$ is continuous and *strongly convex*, i.e., there exists some $K > 0$ such that

$$[\lambda h(x') + (1-\lambda)h(x)] - h(\lambda x' + (1-\lambda)x) \geq \frac{K}{2}\lambda(1-\lambda)\,\|x' - x\|^2 \tag{2.26}$$

for all $x, x' \in \mathcal{X}$ and all $\lambda \in [0,1]$. Moreover, under the empty sum convention $\sum_{s=1}^{0} \ell_s(x) \equiv 0$, the method is initialized at the so-called *prox-center* x_c of $\mathcal{X}$, viz.

$$X_1 = x_c \equiv \operatorname*{arg\,min}_{x\in\mathcal{X}} h(x). \tag{2.27}$$

Remark. We should state here that (FTL) and (FTRL) are closely related to the learning policies known in economics and game theory as *fictitious play* (FP) and *smooth fictitious play* (SFP) respectively. These policies correspond to playing a best response (resp. regularized or smooth best response) to the empirical history of play of one's opponents, and their study dates back to Brown [29], Robinson [121], and Fudenberg and Levine [54]; for a more recent treatment, see also Hofbauer and Sandholm [65].

Blanket assumptions

With all this in hand, the regret analysis of (FTRL) is typically performed under the following blanket assumptions:

Assumption 2.1 (Convexity)**.** Each $\ell_t\colon \mathcal{X} \to \mathbb{R}$ is convex.

Assumption 2.2 (Lipschitz continuity)**.** Each $\ell_t\colon \mathcal{X} \to \mathbb{R}$ is *Lipschitz continuous*, i.e.,

$$|\ell_t(x') - \ell_t(x)| \leq L_t\|x' - x\| \tag{2.28}$$

for some $L_t > 0$ and all $x, x' \in \mathcal{X}$.

Remark. Since ℓ_t is assumed differentiable, Assumption 2.2 basically provides an upper bound for its gradient. In the convex case, all this can be subsumed in the assumption that the subdifferential of ℓ_t on $\mathcal{X}$ admits a bounded selection. We will not require this level of detail at this point, so we stick with the simplest formulation.

In this general setting, we have the following basic result:

Regret of FTRL

Theorem 2.1 (Shalev-Shwartz, 2007)**.** *Suppose that* (FTRL) *is run against a sequence of loss functions ℓ_t, $t = 1, 2, \ldots$, satisfying Assumptions 2.1 and 2.2. Then, the algorithm's regret is bounded by*

$$\mathrm{Reg}(T) \leq \frac{H}{\gamma} + \frac{\gamma}{K}\sum_{t=1}^{T} L_t^2 \tag{2.29}$$

where $H \equiv \max h - \min h$ denotes the "depth" of h over $\mathcal{X}$.[4] *In particular, if $L \equiv \sup_t L_t < \infty$ and* (FTRL) *is run with regularization parameter $\gamma = (1/L)\sqrt{HK/T}$, the incurred regret is bounded as*

$$\mathrm{Reg}(T) \leq 2L\sqrt{(H/K)T}. \tag{2.30}$$

Theorem 2.1 shows that achieving no regret is possible as long as (*i*) the optimizer has full access to the loss functions ℓ_t revealed up to the previous stage (inclusive); (*ii*) the minimization problem defining (FTRL) can be solved efficiently; and (*iii*) the horizon of play is known in advance. Of these limitations, (*iii*) is the easiest to resolve via a method known as the "doubling trick";[5] on the other hand, (*i*) and (*ii*) represent inherent limitations of leader-following policies and are harder to overcome. We address these points in detail in the next section.

4 We write here H instead $H_{\mathcal{X}}$ for notational simplicity.

5 In a nutshell, this involves running (FTRL) over windows of increasing length with a step-size that is chosen optimally for each window; for a detailed account, see Cesa-Bianchi and Lugosi [35], Shalev-Shwartz [136].

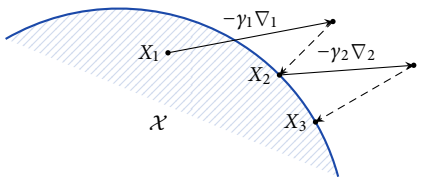


Figure 2.2: Schematic representation of online gradient descent.

Algorithm 2.2: Online gradient descent

Require: step-size sequence $\gamma_t > 0$
choose $X_1 \in \mathcal{X}$ # initialization
for $t = 1, 2, \dots$ **do**
  incur loss $\ell_t(X_t)$ # losses revealed
  receive signal $V_t \leftarrow -[\nabla\ell_t(X_t) + U_t]$ # oracle feedback
  update $X_{t+1} \leftarrow \Pi(X_t + \gamma_t V_t)$ # gradient step
end for

2.2.3 *Online gradient descent*

Online gradient descent

In optimization theory, the most straightforward approach to minimize a given loss function is based on (projected) gradient descent: at each stage, the algorithm takes a step against the gradient of the objective, the resulting point is projected back to the problem's feasible region (if needed), and the process repeats.

When faced with a different loss function at each stage, this gives rise to the policy known as *online gradient descent* (OGD). Formally, this refers to the recursive update rule

$$X_{t+1} = \Pi(X_t + \gamma_t V_t) \tag{OGD}$$

where

$$V_t = -\nabla_t = -[\nabla\ell_t(X_t) + Z_t] \tag{2.31}$$

denotes the return of a stochastic first-order oracle at X_t (cf. Section 2.2.1), $\gamma_t > 0$ is the algorithm's *step-size* (discussed below), and $\Pi\colon \mathcal{V} \to \mathcal{X}$ is the Euclidean projector

$$\Pi(x) = \operatorname*{arg\,min}_{x' \in \mathcal{X}} \|x' - x\|^2. \tag{2.32}$$

[For a schematic representation of the method, see Fig. 2.2; see also Algorithm 2.2 for a pseudocode implementation. For simplicity, we also drop the dependence of Π on $\mathcal{X}$, and we write Π instead of $\Pi_{\mathcal{X}}$.]

Remark 2.1. Before proceeding, it is worth noting a technical discrepancy in (OGD). Specifically, seeing as gradients are formally represented as dual vectors, the addition $X_t + \gamma_t V_t$ of a primal and a dual vector is not well-defined. This issue is usually handwaved away by assuming that $\mathcal{V}$ is a Euclidean (or Hilbert) space, in which case $\mathcal{V}^*$ is canonically identified with $\mathcal{V}$. However, this assumption can only be made if the norm $\|\cdot\|$ satisfies the parallelogram law (i.e., if it is induced by a scalar product); if this is not the case (e.g., if $\|\cdot\|$ is the L^1 norm), the situation is more delicate. We discuss this issue in detail in the next section.

The study of (OGD) in online optimization can be traced back to the seminal paper of Zinkevich [157] who established the following basic bound:

Regret of OGD

Theorem 2.2 (Zinkevich, 2003)**.** *Suppose that* (OGD) *is run against a sequence of loss*

functions satisfying Assumptions 2.1 and 2.2 with a constant step-size $\gamma_t \equiv \gamma > 0$ and SFO feedback of the form (2.31)*. Then, the algorithm's regret is bounded as*

$$\overline{\mathrm{Reg}}(T) \leq \frac{\mathrm{diam}(\mathcal{X})^2}{2\gamma} + \frac{\gamma}{2}\sum_{t=1}^{T} M_t^2 + \mathrm{diam}(\mathcal{X})\sum_{t=1}^{T} B_t \tag{2.33}$$

where $\mathrm{diam}(\mathcal{X}) \equiv \max\{\|x' - x\| : x, x' \in \mathcal{X}\}$ *denotes the diameter of* $\mathcal{X}$*. In particular, if* $M \equiv \sup_t M_t < \infty$ *and* (OGD) *is run with step-size* $\gamma = (1/M)\,\mathrm{diam}(\mathcal{X})/\sqrt{T}$*, the algorithm enjoys the bound*

$$\overline{\mathrm{Reg}}(T) \leq \mathrm{diam}(\mathcal{X})\left[M\sqrt{T} + \sum_{t=1}^{T} B_t\right]. \tag{2.34}$$

Corollary 2.3. *If* (OGD) *is run with unbiased SFO feedback* ($B_t = 0$ *for all* t)*, we have*

$$\overline{\mathrm{Reg}}(T) \leq \mathrm{diam}(\mathcal{X})M\sqrt{T}. \tag{2.35}$$

Finally, if the oracle is perfect ($U_t = 0$ *for all* t)*, the incurred regret is bounded as*

$$\mathrm{Reg}(T) \leq \mathrm{diam}(\mathcal{X})L\sqrt{T}. \tag{2.36}$$

OGD vs. FTRL

Up to a multiplicative constant, the bound (2.36) is essentially the same as the corresponding bound (2.30) for (FTRL); in particular, as long as the oracle does not suffer from systematic errors (or the corresponding bias B_t becomes sufficiently small over time), (OGD) still enjoys an $\mathcal{O}(\sqrt{T})$ regret bound. In other words, (OGD) achieves the same regret minimization rate as (FTRL), even though the latter requires a *full information* oracle. This makes (OGD) significantly more lightweight, so it can be applied to a considerably wider class of problems.

Minimax bounds

We close this section with a brief discussion on the optimality of the bounds (2.35) and (2.36). In this regard, Abernethy et al. [1] showed that an informed adversary choosing linear losses of the form $\ell_t(x) = -\langle v_t, x\rangle$ with $\|v_t\| \leq L$ can impose regret no less than

$$\mathrm{Reg}(T) \geq \frac{\mathrm{diam}(\mathcal{X})L}{2\sqrt{2}}\sqrt{T}. \tag{2.37}$$

This "minimax" bound suggests that there is little hope of improving the regret minimization rate of (OGD) given by (2.33). Nevertheless, despite this negative result, the optimizer can achieve significantly lower regret when facing *strongly convex* losses. More precisely, if each ℓ_t is α-strongly convex (cf. the classification of Section 2.1.1), Hazan et al. [63] showed that (OGD) with a variable step-size of the form $\gamma_t \propto 1/t$ enjoys the *logarithmic* regret guarantee

$$\mathrm{Reg}(T) \leq \frac{1}{2}\frac{L^2}{\alpha}\log T = \mathcal{O}(\log T). \tag{2.38}$$

Logarithmic regret

Importantly, this guarantee is tight in the class of strongly convex functions, even up to the multiplicative constant in (2.38). Specifically, if the adversary is restricted to quadratic convex functions of the form $\ell_t(x) = \frac{1}{2}x^\top A_t x - \langle v_t, x\rangle + c$ with $A_t \succcurlyeq \alpha I$, the optimizer's worst-case regret is bounded from below as

$$\mathrm{Reg}(T) \geq \frac{1}{2}\frac{L^2}{\alpha}\log T. \tag{2.39}$$

This shows that the rate of regret minimization in online convex optimization depends crucially on the curvature of the loss functions encountered. Against arbitrary loss functions, the optimizer cannot hope to do better than $\Omega(\sqrt{T})$; however, if the loss

	MINIMAX REGRET	OGD GUARANTEE
CONVEX	$\Omega(\operatorname{diam}(\mathcal{X})L\sqrt{T})$	$\mathcal{O}(\operatorname{diam}(\mathcal{X})L\sqrt{T})$
α-STRONG	$\Omega(L^2/\alpha \log T)$	$\mathcal{O}(L^2/\alpha \log T)$

Table 2.1: Regret achieved by (OGD) against L-Lipschitz convex losses.

functions encountered possess a uniformly positive global curvature, the optimizer's worst-case guarantee becomes $\mathcal{O}(\log T)$. For convenience, we collect these bounds in Table 2.1.

2.2.4 *Online mirror descent*

The geometry of MABs

Even though the worst-case bound (2.36) for (OGD) is essentially tight, there are cases where the problem's geometry allows for considerably sharper regret guarantees. This is best understood in the MAB setting of Example 2.5: as discussed there, MABs can be seen as online linear optimization problems with action set $\mathcal{X} = \Delta(\mathcal{A}) \equiv \{x \in \mathbb{R}^n : \sum_{a\in\mathcal{A}} x_a = 1\}$ and linear loss functions of the form $\ell_t(x) = -\langle v_t, x\rangle$ for some reward vector $v_t \in \mathbb{R}^n$. The standard payoff normalization assumption for v_t is that $v_{a,t} \in [-1, 1]$ for all $t = 1, 2, \dots$ and all $a \in \mathcal{A}$, so the Lipschitz constant of the bandit's loss functions relative to the Euclidean norm can be bounded by

$$L_2 = \max\{\|v\|_2 : |v_a| \le 1 \text{ for all } a\} = \sqrt{1^2 + \cdots + 1^2} = \sqrt{n}. \tag{2.40}$$

Thus, in view of (2.36), the regret of (OGD) in a MAB problem with perfect oracle feedback is at most

$$\operatorname{Reg}(T) \le 2\sqrt{nT}. \tag{2.41}$$

On the other hand, under the ℓ_∞ norm (i.e., $\|v\|_\infty = \max_{a\in\mathcal{A}}|v_a|$ for $v \in \mathbb{R}^n$), the corresponding Lipschitz constant would be bounded by

$$L_\infty = \max\{\|v\|_\infty : |v_a| \le 1 \text{ for all } a\} = \max_{a\in\mathcal{A}}\{|v_a| : |v_a| \le 1\} = 1. \tag{2.42}$$

Hence, a natural question that arises is whether running (OGD) with a *non-Euclidean* norm can lead to better regret bounds when there are sharper estimates for the Lipschitz constant of the problem's loss functions.[6] This question is at the heart of a general class of online optimization algorithms known collectively as *online mirror descent* (OMD).

OGD revisited

To define it, it will be convenient to rewrite the Euclidean projection in (OGD) in more abstract form as follows: given an input point $x \leftarrow X_t$ and an impulse vector $y \leftarrow \gamma_t V_t$, (OGD) returns the output point $x^+ \leftarrow X_{t+1}$ defined as

$$\begin{aligned} x^+ = \Pi(x + y) &= \operatorname*{arg\,min}_{x'\in\mathcal{X}}\big\{\|x + y - x'\|^2\big\} \\ &= \operatorname*{arg\,min}_{x'\in\mathcal{X}}\big\{\|x - x'\|^2 + \|y\|^2 + 2\langle y, x - x'\rangle\big\} \\ &= \operatorname*{arg\,min}_{x'\in\mathcal{X}}\big\{\langle y, x - x'\rangle + D(x', x)\big\}, \end{aligned} \tag{2.43}$$

where

$$D(x', x) \equiv \tfrac{1}{2}\|x' - x\|^2 = \tfrac{1}{2}\|x'\|^2 - \tfrac{1}{2}\|x\|^2 - \langle x, x' - x\rangle \tag{2.44}$$

The Bregman divergence

denotes the (squared) Euclidean distance between x and x'. Written this way, the basic

6 This is also related to Remark 2.1 on the addition of primal and dual vectors in (OGD). Seeing as the underlying norm now plays an integral part, it is no longer possible to casually identify primal and dual vectors.

idea of mirror descent is to replace this quadratic expression by the more general *Bregman divergence*

$$D(x', x) = h(x') - h(x) - \langle \nabla h(x), x' - x \rangle, \tag{2.45}$$

induced by a "*distance-generating function*" h on $\mathcal{X}$. More precisely, we have:

Distance-generating functions and prox-mappings

Definition 2.2. Let $h: \mathcal{V} \to \mathbb{R} \cup \{\infty\}$ be a proper l.s.c. convex function on $\mathcal{V}$. We say that h is a *distance-generating function* (DGF) on $\mathcal{X}$ if

1. The effective domain of h is $\operatorname{dom} h = \mathcal{X}$.
2. The subdifferential of h admits a *continuous selection*; specifically, writing $\mathcal{X}^\circ \equiv \operatorname{dom} \partial h = \{x \in \mathcal{X} : \partial h(x) \neq \varnothing\}$ for the domain of ∂h, we assume there exists a continuous mapping $\nabla h: \mathcal{X}^\circ \to \mathcal{Y}$ such that $\nabla h(x) \in \partial h(x)$ for all $x \in \mathcal{X}^\circ$.
3. h is K-strongly convex relative to $\|\cdot\|$; in particular

$$h(x') \geq h(x) + \langle \nabla h(x), x' - x \rangle + \frac{K}{2}\|x' - x\|^2 \tag{2.46}$$

for all $x \in \mathcal{X}^\circ$ and all $x' \in \mathcal{X}$.

The *Bregman divergence* $D: \mathcal{X}^\circ \times \mathcal{X} \to \mathbb{R}$ induced by h is then given by Eq. (2.45), and the associated *prox-mapping* $P: \mathcal{X}^\circ \times \mathcal{Y} \to \mathcal{X}$ is defined as

$$P_x(y) = \operatorname*{arg\,min}_{x' \in \mathcal{X}} \{\langle y, x - x' \rangle + D(x', x)\} \quad \text{for all } x \in \mathcal{X}^\circ,\ y \in \mathcal{Y}, \tag{2.47}$$

Remark 2.2. In a slight abuse of notation, when $\mathcal{X}$ is understood from the context, we will not distinguish between h and its restriction $h|_{\mathcal{X}}$ on $\mathcal{X}$.

Remark 2.3. The notion of a distance-generating function is essentially synonymous to that of a regularizer as described in the definition of (FTRL). Regrettably, there is no consensus in the literature regarding terminology and notation: the names "Bregman function" and "link function" are also common for different variants of Definition 2.2. For an entry point to this literature, we refer the reader to Alvarez et al. [4], Beck and Teboulle [12], Bregman [27], Bubeck and Cesa-Bianchi [30], Chen and Teboulle [37], Juditsky et al. [73], Kiwiel [77], Nemirovski et al. [109], Nesterov [110], Shalev-Shwartz [136], and references therein; see also Nemirovski and Yudin [108] for the origins of mirror descent in optimization theory and beyond.

Online mirror descent

With all this in hand, the *online mirror descent* (OMD) policy is defined as

$$X_{t+1} = P_{X_t}(\gamma_t V_t) \tag{OMD}$$

where $\gamma_t > 0$ is a variable step-size sequence, the signals V_t are provided by a stochastic first-order oracle as in (2.31), and P is the prox-mapping induced by some distance-generating function on $\mathcal{X}$. For concreteness, we discuss below two prototypical examples of the method (see also Algorithm 2.3 for a pseudocode presentation):

Euclidean gradient descent

Example 2.6. As we discussed above, the quadratic DGF $h(x) = \frac{1}{2}\|x\|^2$ yields the Euclidean prox-mapping

$$P_x(y) = \operatorname*{arg\,min}_{x' \in \mathcal{X}} \{\langle y, x - x' \rangle + \tfrac{1}{2}\|x' - x\|^2\} = \Pi(x + y). \tag{2.48}$$

Importantly, even though the rightmost side of the above expression involves the addition of a primal and a dual vector, the middle one does not. In view of this, (OMD) is a more natural starting point in non-Euclidean settings where the underlying norm $\|\cdot\|$ is not induced by a scalar product.

Algorithm 2.3: Online mirror descent

Require: strongly convex regularizer h; step-size sequence $\gamma_t > 0$
set $X_1 \leftarrow \arg\min h$ # initialization
for $t = 1, 2, \ldots$ **do**
  incur loss $\ell_t(X_t)$ # losses revealed
  receive signal $V_t \leftarrow -[\nabla \ell_t(X_t) + U_t]$ # oracle feedback
  update $X_{t+1} \leftarrow P_{X_t}(\gamma_t V_t)$ # mirror step
end for

Entropic gradients and exponential weights

Example 2.7. As another example, let $\mathcal{X} \equiv \Delta(\mathcal{A})$ be the standard unit simplex of $\mathbb{R}^n$, and consider the entropic regularizer

$$h(x) = \sum_{a \in \mathcal{A}} x_a \log x_a. \tag{2.49}$$

A standard calculation shows that h is 1-strongly convex relative to the ℓ_1 norm, and the induced prox-mapping may be written as

$$P_x(y) = \frac{(x_a \exp(y_a))_{a \in \mathcal{A}}}{\sum_{a \in \mathcal{A}} x_a \exp(y_a)}. \tag{2.50}$$

We thus obtain the *entropic gradient descent* (EGD) algorithm of Beck and Teboulle [12]:

$$X_{a,t+1} = \frac{X_{a,t} \exp(\gamma_t V_{a,t})}{\sum_{a' \in \mathcal{A}} X_{a',t} \exp(\gamma_t V_{a',t})}. \tag{EGD}$$

In the multi-armed bandit literature (where V_t is a possibly perturbed version of the t-th stage payoff vector v_t), this algorithm is known as *exponential weights* (EW), and it dates back at least to the seminal work of Vovk [150], Littlestone and Warmuth [85], and Auer et al. [7] (see also Example 2.10 below). For a survey, we refer the reader to Arora et al. [6], Bubeck and Cesa-Bianchi [30], and references therein.

We now turn to the basic regret guarantees of (OMD). The main result in this regard is as follows:

Regret of OMD

Theorem 2.4 (Shalev-Shwartz, 2007)**.** *Suppose that* (OMD) *is run against a sequence of loss functions satisfying Assumptions 2.1 and 2.2 with a constant step-size $\gamma_t \equiv \gamma > 0$ and SFO feedback of the form* (2.31)*. Then, the algorithm's regret is bounded as*

$$\overline{\mathrm{Reg}}(T) \leq \frac{H}{\gamma} + \frac{\gamma}{2K} \sum_{t=1}^{T} M_t^2 + \mathrm{diam}(\mathcal{X}) \sum_{t=1}^{T} B_t \tag{2.51}$$

where K is the strong convexity modulus of h and $H \equiv \max h - \min h$ denotes its "depth" over $\mathcal{X}$. In particular, if $M \equiv \sup_t M_t < \infty$ and (OMD) *is run with step-size $\gamma = (1/M)\sqrt{2KH/T}$, the algorithm enjoys the bound*

$$\overline{\mathrm{Reg}}(T) \leq \left[M\sqrt{(2H/K)T} + \mathrm{diam}(\mathcal{X}) \sum_{t=1}^{T} B_t \right]. \tag{2.52}$$

Corollary 2.5. *If* (OMD) *is run with unbiased SFO feedback ($B_t = 0$ for all t), we have*

$$\overline{\mathrm{Reg}}(T) \leq M\sqrt{(2H/K)T}. \tag{2.53}$$

Finally, if the oracle is perfect ($U_t = 0$), the incurred regret is bounded as

$$\mathrm{Reg}(T) \leq L\sqrt{(2H/K)T}. \tag{2.54}$$

Compared to (OGD), the main difference between Theorems 2.2 and 2.4 is the factor $2H/K$ (and, of course, the norm defining L and M). This factor depends on the choice of h in a scale-invariant way (i.e., it remains invariant if h is multiplied by a constant), so finetuning the choice of h to the problem at hand is not always easy. However, in many cases of practical interest, this can be accomplished with remarkable efficiency:

Bandits revisited

Example 2.8. Going back to Example 2.7, the entropic regularizer (2.49) has strong convexity modulus $K = 1$ and its depth over $\mathcal{X} = \Delta(\mathcal{A})$ is

$$H = \max h - \min h = 0 - \sum_{a \in \mathcal{A}} (1/n) \log(1/n) = \log n. \tag{2.55}$$

Hence, if (EGD) is run against a multi-armed bandit with payoffs in $[-1, 1]$ (meaning that $L = 1$ in the ℓ_∞ norm), we obtain the regret bounds

$$\mathrm{Reg}(T) \leq \sqrt{2T \log n} \tag{2.56a}$$

and

$$\overline{\mathrm{Reg}}(T) \leq M\sqrt{2T \log n} \tag{2.56b}$$

corresponding respectively to the perfect and imperfect feedback case (with $\mathbb{E}[V_{a,t}^2 | \mathcal{F}_t] \leq M^2$ for the latter). By comparison, the corresponding bounds for (OGD) are $\mathrm{Reg}(T) \leq 2\sqrt{nT}$ and $\overline{\mathrm{Reg}}(T) \leq 2M\sqrt{nT}$ (for perfect and imperfect feedback respectively), so (EGD) improves on (OGD) by a factor of $\tilde{\Theta}(n)$ in the context of MAB problems.

In short, even though (OMD) enjoys the same $\mathcal{O}(\sqrt{T})$ regret guarantees as (OGD), the multiplicative constants involved may provide a massive improvement relative to the problem's dimension. This is of immense value to real-world machine learning and Big Data problems that suffer from the so-called "curse of dimensionality". As a result, the principled design of tailor-made OMD policies for arbitrary problems has attracted considerable interest in the literature and remains a vigorously researched question.

2.2.5 *Dual averaging and the link between FTRL and OMD*

FTRL and OGD in unconstrained problems

We close this section by discussing an important relation between (FTRL) and (OMD). To see it, consider an unconstrained linear problem with action set $\mathcal{X} = \mathbb{R}^n$, regularization function $h(x) = \frac{1}{2}\|x\|^2$, and linear losses of the form $\ell_t(x) = -\langle v_t, x \rangle$ for some sequence $v_t \in \mathbb{R}^n$. In this case, (FTRL) gives

$$\begin{aligned} X_{t+1} &= \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\{ \sum_{s=1}^{t} \ell_s(x) + \frac{1}{\gamma} h(x) \right\} = \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\{ \|x\|^2 - 2\gamma \sum_{s=1}^{t} \langle v_s, x \rangle \right\} \\ &= \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \left\| x - \gamma \sum_{s=1}^{t} \langle v_s, x \rangle \right\|^2 = \gamma \sum_{s=1}^{t} v_s = \gamma \sum_{s=1}^{t-1} v_s + \gamma v_t = X_t + \gamma v_t \end{aligned} \tag{2.57}$$

i.e., we get the (unprojected) gradient update of (OGD). This is an instance of a much more general link between (FTRL) and (OMD) which we discuss in detail below.

Linearizing FTRL

To begin, we introduce a variant of (FTRL) which only requires first-order oracle information – i.e., the same type of feedback as (OMD). The main idea behind this modification is to replace $\ell_t(x)$ in (FTRL) with the *linear* surrogate

$$\tilde{\ell}_t(x) = \ell_t(X_t) + \langle \nabla \ell_t(X_t), X_t - x \rangle \tag{2.58}$$

which induces the update rule

$$X_{t+1} = \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\{ \sum_{s=1}^{t} \tilde{\ell}_s(x) + \frac{1}{\gamma} h(x) \right\} = \operatorname*{arg\,max}_{x \in \mathcal{X}} \left\{ \gamma \sum_{s=1}^{t} \langle \nabla \ell_s(X_s), x \rangle - h(x) \right\}. \tag{2.59}$$

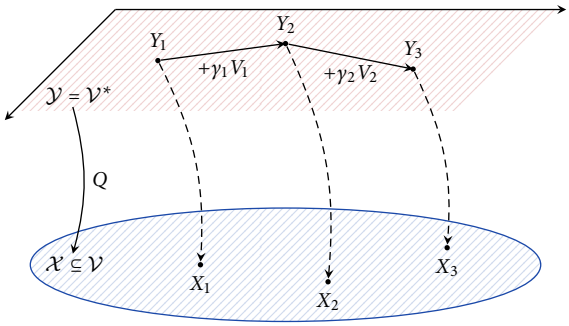


Figure 2.3: Schematic representation of dual averaging.

Algorithm 2.4: Dual averaging

Require: mirror map $Q: \mathcal{Y} \to \mathcal{X}$; step-size sequence $\gamma_t > 0$
choose $Y_1 \in \mathcal{Y}$ # initialization
for $t = 1, 2, \dots$ **do**
  play $X_t \leftarrow Q(Y_t)$ # choose action
  incur loss $\ell_t(X_t)$ # losses revealed
  receive signal $V_t \leftarrow -[\nabla \ell_t(X_t) + U_t]$ # oracle feedback
  update $Y_{t+1} \leftarrow Y_t + \gamma_t V_t$ # dual step
end for

In contrast to (FTRL), this policy only requires first-order information on ℓ_t (and, of course, coincides with (FTRL) in the case of linear losses). Taking this a step further, if the feedback available to the optimizer is a gradient signal V_t of the form (2.31), we obtain the *follow-the-linearized-leader* (FTLL) policy

$$X_{t+1} = \operatorname*{arg\,max}_{x \in \mathcal{X}} \left\{ \gamma \sum_{s=1}^{t} V_s - h(x) \right\}. \tag{FTLL}$$

Dual averaging

In turn, written in recursive form, (FTLL) yields the *dual averaging* (DA) method

$$\begin{aligned} Y_{t+1} &= Y_t + \gamma_t V_t \\ X_{t+1} &= Q(Y_{t+1}) \end{aligned} \tag{DA}$$

where:

1. $Y_t \in \mathcal{Y}$ is an auxiliary dual variable that aggregates gradient steps.
2. $\gamma_t > 0$ is a (variable) step-size parameter.
3. $Q: \mathcal{Y} \to \mathcal{X}$ is the so-called "mirror map" of h and is defined as

$$Q(y) = \arg\max\{\langle y, x \rangle - h(x)\} \quad \text{for all } y \in \mathcal{Y}. \tag{2.60}$$

The terminology "dual averaging" is due to Nesterov [111] and alludes to the fact that gradients are "averaged" in the dual space $\mathcal{Y} \equiv \mathcal{V}^*$ (where they belong) before being "mirrorred" back to the problem's feasible region $\mathcal{X} \subseteq \mathcal{V}$. By contrast, in the online learning literature, (DA) is often referred to as the "lazy" variant of OGD/OMD [136, 157].[7] For concreteness, we provide some key examples below (see also Fig. 2.3 for a schematic representation and Algorithm 2.4 for a pseudocode implementation):

Lazy gradient descent

Example 2.9. Returning to the Euclidean framework of Example 2.6, the quadratic

7 In this context, the original incarnation of OGD/OMD is referred to as "greedy" or "eager".

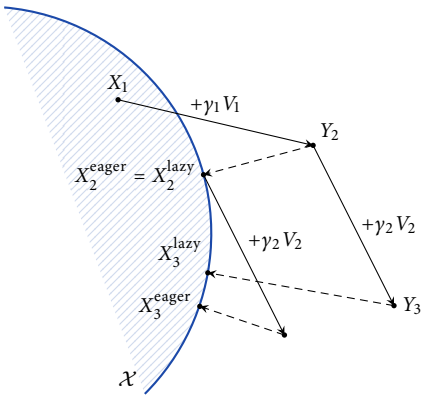


Figure 2.4: Lazy vs. eager gradient descent.

regularizer $h(x) = \frac{1}{2}\|x\|^2$ on $\mathcal{X} \in \mathbb{R}^n$ yields the mirror map

$$Q(y) = \operatorname*{arg\,max}_{x\in\mathcal{X}}\{\langle y, x\rangle - \tfrac{1}{2}\|x\|^2\} = \operatorname*{arg\,min}_{x\in\mathcal{X}}\|x - y\|^2 = \Pi(y) \tag{2.61}$$

where $\Pi(y)$ denotes the Euclidean projection (2.32) of $y \in \mathbb{R}^n$ onto $\mathcal{X}$. In this way, we obtain the *lazy gradient descent* (LGD) policy:

$$\begin{aligned} Y_{t+1} &= Y_t + \gamma_t V_t \\ X_{t+1} &= \Pi(Y_{t+1}) \end{aligned} \tag{LGD}$$

The adjective "lazy" refers to the fact that the algorithm aggregates gradient steps "lazily" (i.e., without transporting them to the state at which they were generated), and only projects to $\mathcal{X}$ in order to generate a new gradient signal. In view of this, (LGD) agrees with (OGD) when $\mathcal{X}$ is an affine subspace of $\mathbb{R}^n$, but not otherwise; we illustrate the difference between the two algorithms in Fig. 2.4.

Hedge and exponential weights

Example 2.10. Going back to Example 2.7, a straightforward calculation shows that the mirror map associated to the entropic regularizer $h(x) = \sum_{a\in\mathcal{A}} x_a \log x_a$ over the unit simplex $\mathcal{X} = \Delta(\mathcal{A})$ is the *logit choice* map

$$\Lambda(y) = \frac{(\exp(y_1), \dots, \exp(y_n))}{\exp(y_1) + \cdots + \exp(y_n)} \tag{2.62}$$

which, in turn, leads to the so-called "*Hedge*" policy[8]

$$\begin{aligned} Y_{t+1} &= Y_t + \gamma_t V_t \\ X_{t+1} &= \Lambda(Y_{t+1}) \end{aligned} \tag{Hedge}$$

Unfolding the above, we readily get

$$X_{a,t+1} \propto \exp(Y_{a,t} + \gamma_t V_{a,t}) \propto X_{a,t}\exp(V_{a,t}) \tag{2.63}$$

which, given the constant normalization $\sum_a X_{a,t} = 1$, implies that the sequence of iterates of (Hedge) is the same as (EGD). In other words, in the case of entropic regularization, the "lazy" and "eager" variants of (OMD) coincide.

8 The terminology "Hedge" is due to Auer et al. [7]. The algorithm is also referred to as the *exponential weights* (EW) or *multiplicative weights* (MW) algorithm [6, 35].

The two examples above suggest that the choice of distance-generating function plays an important role in determining the link between (DA) and (OMD). As we show below, the precise relation is determined by the subdifferentiability of h:

When eager and lazy schemes coincide

Proposition 2.6. *Consider the eager and lazy update rules*

$$x^{\text{eager}} = P_x(w) \tag{2.64a}$$

$$x^{\text{lazy}} = Q(y + w) \tag{2.64b}$$

where the impulse vector $w \in \mathcal{Y}^*$ is arbitrary and, for consistency, the initial points $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ satisfy $x = Q(y)$. Suppose further that $y - \nabla h(x)$ annihilates all tangent vectors to $\mathcal{X}$ at x, i.e.,

$$\langle y - \nabla h(x), x' - x \rangle = 0 \quad \text{for all } x' \in \mathcal{X}. \tag{2.65}$$

Then, $x^{\text{eager}} = x^{\text{lazy}}$.

Corollary 2.7. *If $\mathcal{X}^\circ \equiv \operatorname{dom} \partial h = \operatorname{ri}(\mathcal{X})$, then $x^{\text{eager}} = x^{\text{lazy}}$ for all $x \in \mathcal{X}^\circ$.*

Proof. By the definitions (2.47) and (2.60) of P and Q respectively, we have:

$$\begin{aligned} P_x(w) &= \operatorname*{arg\,min}_{x' \in \mathcal{X}} \{\langle w, x - x' \rangle + D(x', x)\} \\ &= \operatorname*{arg\,min}_{x' \in \mathcal{X}} \{\langle w, x - x' \rangle + h(x') - h(x) - \langle \nabla h(x), x' - x \rangle\} \\ &= \operatorname*{arg\,min}_{x' \in \mathcal{X}} \{\langle w, x - x' \rangle + h(x') - h(x) - \langle y, x' - x \rangle\} \\ &= \operatorname*{arg\,max}_{x' \in \mathcal{X}} \{\langle y + w, x' \rangle - h(x')\} = Q(y + w) \end{aligned} \tag{2.66}$$

i.e., $x^{\text{eager}} = x^{\text{lazy}}$, as claimed. □

Steepness

Heuristically, Proposition 2.6 implies that the lazy and eager variants of (OMD) coincide as long as h becomes "steep" at the boundary of $\mathcal{X}$, i.e.,

$$\lim_{k \to \infty} \|\nabla h(x_k)\|_* = \infty \quad \text{whenever} \quad \lim_{k \to \infty} x_k \in \operatorname{bd}(\mathcal{X}). \tag{2.67}$$

The two prototypical examples discussed above illustrate this dichotomy particularly well: in the Euclidean case, the inclusion $\operatorname{ri}(\mathcal{X}) \subseteq \operatorname{dom} \partial h$ is proper (at least when $\mathcal{X}$ is not an affine space), so the lazy and eager variants of (OMD) are different; by contrast, the entropic regularizer has $\operatorname{ri}(\mathcal{X}) = \operatorname{dom} \partial h$, so the lazy and eager variants coincide.

We close this section by noting that, irrespective of steepness, the lazy and eager variants of (OMD) enjoy the same regret bounds. Specifically:

Regret of DA

Theorem 2.8. *Suppose that* (DA) *is run against a sequence of loss functions satisfying Assumptions 2.1 and 2.2 with a constant step-size γ and SFO feedback of the form* (2.31)*. Then, the incurred regret satisfies* (2.51) *and Corollary 2.5 applies.*

On account of the above, we will make little distinction between the lazy and eager variants of (OMD) and we will frequently drop the lazy/eager characterization altogether. In particular, in view of the optimal no-regret properties of mirror descent methods, we will often treat them as synonymous with "no-regret learning".

2.3 THE MULTI-AGENT VIEWPOINT: GAMES AND EQUILIBRIUM

Up to this point, the environment generating the agent's rewards was treated as an abstract "adversary", a black box with no individual stake in the game. In this section, we take a more detailed look at multi-agent learning – and, specifically, *learning in games*.

2.3.1 *Basic definitions and examples*

Non-cooperative games

In its most basic form, a "game" is a mathematical framework for modeling strategic interactions between optimizing agents with different individual objectives – the *players* of the game. Of course, there is an extensive taxonomy of game-theoretic models depending on the type of players involved (e.g., atomic vs. non-atomic), the nature of the players' interactions (whether they are sequential or simultaneous, cooperative or non-cooperative), and/or the way these interactions determine the agents' rewards. However, there are three principal components that are present throughout this diverse landscape: *i*) the *players* of the game; *ii*) each player's set of *actions*; and *iii*) the players' *payoff functions*. Specifying these three elements goes a long way in defining the relevant solution concepts and what may or may not be learnable in this context.

Continuous games

Building on the online optimization framework of the previous sections, we will focus almost exclusively on non-cooperative games with a finite number of players and continuous action spaces; in the spirit of Debreu [48], such games will be called *continuous games*. Concretely, in a continuous game, players are indexed by a finite set $\mathcal{N} = \{1, \dots, N\}$, and every player $i \in \mathcal{N}$ is assumed to select an action x_i from a compact convex subset $\mathcal{X}_i$ of an ambient, finite-dimensional space $\mathcal{V}_i \cong \mathbb{R}^{n_i}$. The reward of the i-th player is then determined by their individual action and the action $x_{-i} \equiv (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N)$ of the player's opponents.[9]

More precisely, writing $\mathcal{X} \equiv \prod_{i\in\mathcal{N}} \mathcal{X}_i$ for the game's action space and $\mathcal{V} \equiv \prod_{i\in\mathcal{N}} \mathcal{V}_i$ for the corresponding ambient space, we assume that each player's reward is determined by a continuous *payoff* (or *utility*) *function* $u_i \colon \mathcal{X} \to \mathbb{R}$ which maps an action profile $x = (x_i; x_{-i}) \equiv (x_1, \dots, x_N) \in \mathcal{X}$ to its associated reward $u_i(x)$. A *continuous game* will then be a tuple $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ with players, action spaces and payoffs defined as above.

We describe below some important examples of continuous games:

Finite games and their mixed extensions

Example 2.11. In a *finite game*, each player $i \in \mathcal{N}$ chooses a *pure strategy* a_i from a finite set $\mathcal{A}_i$. The players' payoffs are then determined by the pure strategy profile $a = (a_1, \dots, a_N)$ and a collection of payoff functions $u_i \colon \mathcal{A} \equiv \prod_j \mathcal{A}_j \to \mathbb{R}$, $i = 1, \dots, N$. In the *mixed extension* of a finite game, players are allowed to randomize their decisions by playing *mixed strategies*, i.e., probability distributions $x_i = (x_{ia_i})_{a_i\in\mathcal{A}_i} \in \Delta(\mathcal{A}_i)$ with the interpretation that x_{ia_i} represents the probability of choosing action $a_i \in \mathcal{A}_i$ (i.e., as in multi-armed bandits). In this case (and in a slight abuse of notation), the expected payoff to player i under the mixed strategy profile $x \equiv (x_i; x_{-i}) = (x_1, \dots, x_N)$ is

$$u_i(x_i; x_{-i}) = \sum_{a_1\in\mathcal{A}_1} \cdots \sum_{a_N\in\mathcal{A}_N} u_i(a_1, \dots, a_N)\, x_{1,a_1} \cdots x_{N,a_N}. \tag{2.68}$$

Since each player's mixed strategy space $\mathcal{X}_i = \Delta(\mathcal{A}_i)$ is convex and u_i is individually linear in x_i, mixed extensions of finite games are obviously continuous games. When we need to distinguish between a finite game and its mixed extension, we will write $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ for the former and $\Delta(\Gamma)$ for the latter.

Zero-sum games

Example 2.12. Consider a saddle-point problem of the general form

$$\min_{x_1\in\mathcal{X}_1} \max_{x_2\in\mathcal{X}_2} \Phi(x_1, x_2) \tag{SP}$$

where each $\mathcal{X}_i$, $i = 1, 2$, is a compact convex subset of $\mathbb{R}^{n_i}$ and the value function $\Phi \colon \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ is assumed jointly continuous. This problem is the protypical example of a *zero-sum game* with player set $\mathcal{N} = \{1, 2\}$ and payoff functions $u_1 = -\Phi$ and $u_2 = \Phi$. Zero-sum games of this type have been at the core of game-theoretic research from its

9 We use here the notation "$-i$" for the family of indices $\mathcal{N}_{-i} \equiv \mathcal{N} \setminus \{i\}$.

earliest steps in the late 1920's [149] to its most recent and successful applications in machine learning and artificial intelligence [56].

Kelly auctions

Example 2.13. Consider a service provider with a *splittable good* that is to be auctioned off (bandwidth, ad display time, etc.). Specifically, fractions of this good can be leased to N bidders (players) who can place monetary bids $x_i \geq 0$ for each good up to their total budget b_i. Once all bids are in, the good is sliced out proportionally to each player's bid, with the i-th player getting $\rho_i = x_i \big/ (c + \sum_{j \in \mathcal{N}} x_j)$ units of the auctioned good (where $c \geq 0$ represents an "entry barrier" for bidding). A simple model for the utility of player i is then given by

$$u_i(x_i; x_{-i}) = g_i \rho_i - x_i, \tag{2.69}$$

with g_i denoting the marginal gain of player i from acquiring a unit of goods. Writing $\mathcal{X}_i = [0, b_i]$ for the action space of player i, the game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ is known as the *Kelly auction* [75] and is one of the principal models for auctions with splittable goods.[10]

Cournot competition

Example 2.14. Consider a finite set $\mathcal{N} = \{1, \dots, N\}$ of *firms*, each supplying the market with a quantity $x_i \in [0, C_i]$ of some good (or service) up to the firm's production capacity C_i. This good is then priced as a decreasing function $P(x)$ of each firm's production; for concreteness, we focus on the linear model $P(x) = a - \sum_i b_i x_i$ where a is a positive constant and the coefficients $b_i > 0$ reflect the price-setting power of each firm.

In this model, the utility of firm i is given by

$$u_i(x) = x_i P(x) - c_i x_i, \tag{2.70}$$

where c_i represents the marginal production cost of firm i. Switching from costs to rewards, the resulting game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, -c)$ is known as a *Cournot competition game* and plays a central role in oligopoly theory.

Congestion games

Example 2.15. Congestion games are game-theoretic models that arise in the study of traffic networks (such as the Internet) [17, 125, 129]. To define them, fix a set of players $\mathcal{N} = \{1, \dots, N\}$ that share a set of *resources* $r \in \mathcal{R}$, with each resource associated to a nondecreasing convex *cost function* $c_r \colon \mathbb{R}_+ \to \mathbb{R}$ (for instance, links in a data network and their corresponding delay functions). Each player $i \in \mathcal{N}$ has a certain *traffic demand* $\rho_i > 0$ which is split over a collection $\mathcal{P}_i \subseteq 2^{\mathcal{R}}$ of resource subsets p_i of $\mathcal{R}$ – e.g., sets of links that form origin-destination paths in the network.

In this setting, the action space of each player $i \in \mathcal{N}$ is defined as the scaled simplex $\mathcal{X}_i = \rho_i \, \Delta(\mathcal{P}_i) = \{x_i \in \mathbb{R}_+^{\mathcal{P}_i} : \sum_{p_i \in \mathcal{P}_i} x_{i,p_i} = \rho_i\}$ of *load distributions* over $\mathcal{P}_i$. Then, given a load profile $x = (x_1, \dots, x_N)$, costs are determined based on the utilization of each resource as follows: First, the *demand* w_r of the r-th resource is defined as the total load $w_r = \sum_{i \in \mathcal{N}} \sum_{p_i \ni r} x_{i,p_i}$ on said resource. This demand incurs a *cost* $c_r(w_r)$ per unit of load to each player utilizing resource r, where $c_r \colon \mathbb{R}_+ \to \mathbb{R}$ is a nondecreasing convex function. Accordingly, the total cost to player $i \in \mathcal{N}$ is

$$c_i(x) = \sum_{p_i \in \mathcal{P}_i} x_{i,p_i} c_{i,p_i}(x), \tag{2.71}$$

where $c_{i,p_i}(x) = \sum_{r \in p_i} c_r(w_r)$ is the cost incurred to player i by the utilization of $p_i \in \mathcal{P}_i$. Switching again from costs to rewards, the resulting N-person continuous game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, -c)$ is called an *atomic splittable congestion game*.

10 E. Altman pointed out to me that the class of rent-seeking games considered by Tullock [146] is essentially equivalent to the Kelly auction described above. However, seeing as the motivation of rent-seeking games is not related to the setting in hand, we will use the term "Kelly mechanism" throughout.

2.3.2 *Nash equilibrium*

The most prevalent solution concept in game theory is that of a Nash equilibrium, defined here as an action profile $x^* \in \mathcal{X}$ that is resilient to unilateral deviations. More formally, we have the following definition:

Nash equilibrium

Definition 2.3. An action profile $x^* \in \mathcal{X}$ is said to be a *Nash equilibrium* (NE) of the continuous game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ if

$$u_i(x_i^*; x_{-i}^*) \geq u_i(x_i; x_{-i}^*) \quad \text{for all } x_i \in \mathcal{X}_i \text{ and all } i \in \mathcal{N}. \tag{NE}$$

If (NE) holds as a strict inequality for all deviations $x_i \neq x_i^*$ and all players $i \in \mathcal{N}$, we will say that x^* is a *strict* Nash equilibrium. The set of Nash equilibria of $\mathcal{G}$ will be denoted throughout as $\mathcal{X}^* \equiv \mathrm{NE}(\mathcal{G})$.

Existence

The existence of equilibria in the class of convex-concave zero-sum games (i.e., the setting of Example 2.12 with Φ convex-concave) was first discussed by von Neumann (1928). The second groundbreaking equilibrium existence result (and the namesake of the concept) is due to Nash (1951), who showed that any *finite* game admits an equilibrium in mixed strategies (or, more formally, that the mixed extension of any finite game admits a Nash equilibrium). Finally, unifying these two existence results, Debreu (1952) showed that any continuous game with compact action spaces admits a Nash equilibrium, provided the following individual concavity assumption holds:[11]

$$u_i(x_i; x_{-i}) \text{ is concave in } x_i \text{ for all } x_{-i} \in \mathcal{X}_{-i},\ i \in \mathcal{N}. \tag{2.72}$$

Concave games

Following Rosen [122], games satisfying the individual concavity assumption (2.72) are called *concave games*. The class of concave games is particularly rich and has a broad range of applications in signal processing, wireless communications, economics, and many other disciplines [91, 113, 134]. In particular, a quick check reveals that Examples 2.11–2.15 are all concave (provided that Φ is convex-concave in Example 2.12).

Variational characterization

In concave games with smooth payoff functions,[12] Nash equilibria can also be characterized via the first-order optimality condition

$$\langle V_i(x^*), x_i - x_i^* \rangle \leq 0 \quad \text{for all } x_i \in \mathcal{X}_i,\ i \in \mathcal{N}, \tag{2.73}$$

where $V_i(x)$ denotes the (negative) *individual payoff gradient* of the i-th player,i.e.,

$$V_i(x) = \nabla_{x_i} u_i(x_i; x_{-i}), \tag{2.74}$$

and ∇_{x_i} denotes differentiation with respect to the variable x_i. This variational characterization of Nash equilibria can be written more concisely (but otherwise equivalently) as a variational inequality of the form

$$\langle V(x^*), x - x^* \rangle \leq 0 \quad \text{for all } x \in \mathcal{X} \tag{VI}$$

where

$$V(x) = (V_1(x), \ldots, V_N(x)) \tag{2.75}$$

11 In fact, Debreu [48] only required quasi-concavity and considered the case where the feasible actions of a given player may depend on the actions chosen by another player. We will not work at this level of generality.

12 There is a recent tendency in machine learning to refer to such games as *differentiable games* – see e.g., Balduzzi et al. [10] and many of the references therein. Given the extensive literature on *differential* games (which contains pursuit-evasion and mean-field models), this terminology is, at best, unfortunate. Given the work of [49] on smooth preference models, the adjective "smooth" seems more appropriate in this context. This is the terminology used by Laraki et al. [83], but it conflicts in turn with the concept of "(λ, μ)-smoothness" for finite games; mathematically, I find "smoothness" to be a meaningless term in finite games, but its use is fairly entrenched in the algorithmic game theory literature.

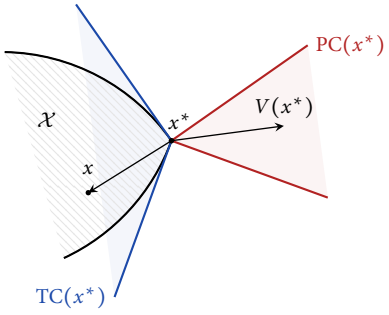


Figure 2.5: Variational characterization of Nash equilibria in concave games.

denotes the players' individual payoff gradient profile at $x \in \mathcal{X}$. By contrast, if $\mathcal{G}$ is not concave, (VI) only determines the game's *critical points*, i.e., those profiles for which an $\mathcal{O}(\varepsilon)$ unilateral deviation cannot increase the payoff of the deviating player by more than $\mathcal{O}(\varepsilon^2)$. [For a schematic illustration, see Fig. 2.5.]

Example 2.16. As an example, in the context of finite games (cf. Example 2.11), the players' individual payoff gradient field can be written in components as

$$\begin{aligned} V_{ia_i}(x) = \frac{\partial u_i}{\partial x_{ia_i}} &= \sum_{a_1' \in \mathcal{A}_1} \cdots \sum_{a_N' \in \mathcal{A}_N} x_{1,a_1'} \cdots \delta_{a_i,a_i'} \cdots x_{N,a_N'}\; u_i(a_1', \dots, a_N') \\ &= u_i(a_i; x_{-i}) \equiv u_{ia_i}(x). \end{aligned} \tag{2.76}$$

In other words, $V_{ia_i}(x)$ is simply the payoff $u_{ia_i}(x) \equiv u_i(a_i; x_{-i})$ to player i when they select $a_i \in \mathcal{A}_i$ against the mixed strategy profile $x_{-i} \in \mathcal{X}_{-i}$ of i's opponents.

Uniqueness

In terms of equilibrium uniqueness, Rosen [122] used a variational characterization similar to (VI) to establish the following sufficient condition:

Theorem 2.9 (Rosen, 1965)**.** *Assume that $\mathcal{G}$ satisfies the payoff monotonicity condition*

$$\langle V(x') - V(x), x' - x \rangle \leq 0 \quad \text{for all } x, x' \in \mathcal{X}, \tag{MC}$$

with equality if and only if $x = x'$. Then, $\mathcal{G}$ admits a unique Nash equilibrium.

Monotone games

Owing to the link between (MC) and the theory of monotone operators, we will refer to games satisfying (MC) as *monotone games*.[13] More precisely, mirroring the corresponding terminology from operator theory, we will say that a game is:

1. *Monotone* if it satisfies (MC).

2. *Strictly monotone* if (MC) holds as a strict inequality whenever $x' \neq x$.

3. *Strongly monotone* if there exists a positive constant $\alpha > 0$ such that

$$\langle V(x') - V(x), x' - x \rangle \leq -\alpha \|x' - x\|^2 \quad \text{for all } x, x' \in \mathcal{X}. \tag{2.77}$$

13 Rosen [122] uses the name *diagonal strict concavity* (DSC) for a weighted variant of (MC) which holds as a strict inequality when $x' \neq x$. Hofbauer and Sandholm [65] use the term "stable" to refer to a class of population games that satisfy a condition similar to (MC), while Sandholm [130] and Sorin and Wan [140] respectively call such games "contractive" and "dissipative". We use the term "monotone" throughout to underline the connection of (MC) with operator theory and variational inequalities (though operator monotonicity is usually defined with the opposite sign to be consistent with function minimization in convex optimization).

Obviously, "strongly monotone" $\subsetneq$ "strictly monotone" $\subsetneq$ "monotone", just as in the chain of inclusions "strongly convex" $\subsetneq$ "strictly convex" $\subsetneq$ "convex" for convex functions. Moreover, letting $x'_{-i} = x_{-i}$ in (MC), we get

$$\langle V_i(x'_i; x_{-i}) - V_i(x_i; x_{-i}), x'_i - x_i \rangle \leq 0 \quad \text{for all } x_i, x'_i \in \mathcal{X}_i,\ x_{-i} \in \mathcal{X}_{-i}. \tag{2.78}$$

This means that the individual payoff gradient $V_i(x)$ of the i-th player is itself monotone in x_i, implying in turn that $u_i(x)$ is concave in x_i for all i [11]. Hence, any game satisfying (MC) is a fortiori concave.

In regard to the examples discussed earlier in this section, Example 2.12 is monotone when Φ is convex-concave; the games in Examples 2.13 and 2.14 are both strictly monotone; [14] and the atomic splittable congestion games of Example 2.15 are also monotone in networks with parallel links and convex latency functions [115]. Together with Rosen's uniqueness theorem,[15] the wide variety of applications in which (MC) holds makes the class of monotone games a particularly rich and interesting one.

2.3.3 *Correlated and coarse correlated equilibrium*

CORRELATED EQUILIBRIUM. A common critique of Nash equilibrium is that a player has no incentive to commit to their component of an equilibrium strategy unless all other players are also expected to play theirs. This argument gains additional momentum if the game in question has multiple Nash equilibria: in that case, even players with unbounded deductive capabilities would be hard-pressed to choose a strategy. This point of view led Aumann [8, 9] to introduce the notion of a *correlated equilibrium* (CE), where subjective beliefs are also taken into account.[16]

Correlation and subjective beliefs

For simplicity, we will define correlated equilibria in the context of finite games where complicated measurability issues do not arise (what this means will become clear below). To do so, let $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ be a finite game in normal form (cf. Example 2.11) and, following Aumann [9], assume that the players' beliefs are formed by observing the "state of the world", i.e., an event drawn from some (finite) probability space $(\Omega, \mathbb{P})$.[17] This data is observed, recorded and processed by the players, who then choose an action based on their individual – but otherwise *correlated* – assessment of their surroundings.

Correlated strategies

More formally, define a *correlated strategy* as a map $\pi\colon \Omega \to \mathcal{A} \equiv \prod_i \mathcal{A}_i$ whose components $\pi_i\colon \Omega \to \mathcal{A}_i$ determine the response $\pi_i(\omega) \in \mathcal{A}_i$ of the i-th player when the world is at state $\omega \in \Omega$. Then, if we write

$$\chi_a \equiv \chi_{a_1,\dots,a_N} = \mathbb{P}\{\omega : \pi(\omega) = a\}. \tag{2.79}$$

14 For a proof in the case of Kelly auctions, see Bravo et al. [26]; for Cournot oligopolies, monotonicity follows from the fact that the game admits a concave potential in the sense of Monderer and Shapley [103].

15 An immediate generalization of Theorem 2.9 is that the set of Nash equilibria of a monotone game is convex and compact even if the game is not *strictly* monotone (in which case its Nash set is a singleton). All in all, the link between variational inequalities and Nash equilibria has given rise to an extensive literature at the interface of game theory and optimization; for an overview, we refer the reader to Facchinei and Pang [50], Mertikopoulos and Zhou [97], and references therein.

16 A few years later, Brian Arthur [28] put forth another argument for the use of correlated equilibrium as a predictive tool: while humans are only moderately strong in problems that can be solved by *deductive* reasoning (they do better than animals but much worse than computers), they excel in intuition and in solving problems by *inductive* reasoning. Since this "intuitive" approach rests heavily on what players believe is going on around them, an equilibrium is only reachable if it also takes into account these beliefs.

17 Of course, this brings up the issue of exactly what kind of information is actually observable by a player. To account for that, Aumann partitions Ω in player-specific σ-algebras which determine whether an event is "observable" (measurable) by a player or not. To keep our discussion as simple as possible, we will not concern ourselves with this issue here.

for the probability of observing the profile $a = (a_1, \dots, a_N) \in \mathcal{A}$, a correlated strategy can also be viewed as an element of the simplex $\mathcal{X}_c \equiv \Delta\left(\prod_i \mathcal{A}_i\right) = \{x \in \mathbb{R}_+^{\mathcal{A}} : \sum_{a\in\mathcal{A}} x_a = 1\}$. On that account, we will interchangeably refer to both π and χ as a correlated strategy, and we will only distinguish between the two when there is danger of confusion.

Clearly, the space of correlated strategies contains the simplex $\mathcal{X} \equiv \prod_i \Delta(\mathcal{A}_i)$ of *mixed* strategies as the subset of *uncorrelated* strategies. Specifically, if we denote the marginals of χ as

$$x_{ia_i} \equiv \mathbb{P}(\pi_i = a_i) \tag{2.80}$$

the condition $\chi_a = \prod_i x_{ia_i}$ which characterizes mixed srategies holds if and only if the individual components of π are stochastically independent (viewed here as random variables in their own right). Thus, under a slight abuse of notation, the expected payoff to the i-th player in a correlated strategy $\chi \in \mathcal{X}_c$ may be written as

$$u_i(\pi) = \sum_{a\in\mathcal{A}} \mathbb{P}\{\omega : \pi(\omega) = a\}\, u_i(a) = \sum_{a_1\in\mathcal{A}_1} \cdots \sum_{a_N\in\mathcal{A}_N} \chi_{a_1,\dots,a_N}\, u_i(a_1, \dots, a_N). \tag{2.81}$$

Heuristically, a correlated strategy π may be viewed as a "coordination device" that outputs a specific recommendation $\pi_i(\omega)$ to each player $i \in \mathcal{N}$ when the state of the world is $\omega \in \Omega$. The notion of a correlated equilibrium posits that a player has no incentive to deviate from this recommendation when averaging over all states:

Correlated equilibrium

Definition 2.4. A correlated strategy π^* is a *correlated equilibrium* (CE) of the finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ if

$$u_i(\pi^*) \geq u_i(\pi_i; \pi^*_{-i}) \tag{CE}$$

for all players $i \in \mathcal{N}$ and all strategies $\pi_i : \Omega \to \mathcal{A}_i$ that factor through π_i^* (i.e., $\pi_i = \sigma_i \circ \pi_i^*$ for some strategy modification $\sigma_i : \mathcal{A}_i \to \mathcal{A}_i$). The set of correlated equilibria of Γ will be denoted throughout as $\mathcal{X}_c^* \equiv \mathrm{CE}(\Gamma)$.

At first sight, the factoring requirement might appear artificial, but it is a vital ingredient of the definition of correlated strategies. Indeed, a given player $i \in \mathcal{N}$ may either follow the recommendation $\pi_i^*(\omega)$ of an equilibrium strategy π_i^*, or disregard it altogether and play something different. However, since the only information that reaches the player in this picture is the recommendation $\pi_i^*(\omega)$ (and not the actual state ω), the player's action may only depend on $\pi_i^*(\omega)$, i.e., be of the form $\sigma_i(\pi_i^*(\omega))$ for some endomorphism $\sigma_i : \mathcal{A}_i \to \mathcal{A}_i$.

Probabilistic characterization

An alternative characterization of (CE) which highlights precisely this feature of correlated equilibria is obtained by the simple rearrangement:

$$\begin{aligned} u_i(\pi_i; \pi^*_{-i}) &= \sum_{a_i\in\mathcal{A}_i} \sum_{a_{-i}\in\mathcal{A}_{-i}} \mathbb{P}(\pi_i = a_i; \pi^*_{-i} = a_{-i})\, u_i(a_i; a_{-i}) \\ &= \sum_{a_i\in\mathcal{A}_i} \sum_{a_{-i}\in\mathcal{A}_{-i}} \left(\sum_{a_i' : \sigma_i(a_i') = a_i} \mathbb{P}(\pi_i^* = a_i; \pi^*_{-i} = a_{-i}) \right) u_i(a_i; a_{-i}) \\ &= \sum_{a_i\in\mathcal{A}_i} \sum_{a_{-i}\in\mathcal{A}_{-i}} \chi^*_{a_i;a_{-i}}\, u_i(\sigma_i(a_i); a_{-i}). \end{aligned} \tag{2.82}$$

The correlated equilibrium condition (CE) may thus be rewritten as:

$$\sum_{a_i\in\mathcal{A}_i} \sum_{a_{-i}\in\mathcal{A}_{-i}} \chi^*_{a_i;a_{-i}} \left[u_i(a_i; a_{-i}) - u_i(a_{-i}; \sigma_i(a_i))\right] \geq 0 \tag{2.83}$$

for all players $i \in \mathcal{N}$ and all maps $\sigma_i : \mathcal{A}_i \to \mathcal{A}_i$. More explicitly, we have:

Proposition 2.10 (Aumann, 1987)**.** *A correlated strategy $\chi^* \in \mathcal{X}_c$ is a correlated equilibrium of a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ if and only if*

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} \chi^*_{a_i; a_{-i}} u_i(a_i; a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} \chi^*_{a_i; a_{-i}} u_i(a'_i; a_{-i}) \tag{2.84}$$

for all $i \in \mathcal{N}$ and all $a_i, a'_i \in \mathcal{A}_i$.

Remark 2.4. If $x^*_{ia_i} > 0$, we may divide both sides of (2.84) by $x^*_{ia_i}$ to obtain the conditional version:

$$\sum_{a_{-i} \in \mathcal{A}_{-i}} \mathbb{P}(\pi^*_{-i} = a_{-i} | \pi^*_i = a_i) u_i(a_i; a_{-i}) \geq \sum_{a_{-i} \in \mathcal{A}_{-i}} \mathbb{P}(\pi^*_{-i} = a_{-i} | \pi^*_i = a_i)\, u_i(a'_i; a_{-i}) \tag{2.85}$$

where, in obvious notation, $\mathbb{P}(\pi^*_{-i} = a_{-i} \mid \pi^*_i = a_i)$ denotes the conditional probability

$$\mathbb{P}(\pi^*_{-i} = a_{-i} \mid \pi^*_i = a_i) = \frac{\mathbb{P}(\pi^* = a)}{\mathbb{P}(\pi^*_i = a_i)}. \tag{2.86}$$

This last form of (2.84) highlights even more clearly the idea of deviating from an equilibrium recommendation.

Nash vs. correlated equilibria

Algebraically, Proposition 2.10 shows that correlated equilibria can be computed efficiently by solving a system of linear inequalities (i.e., by solving a linear problem). By contrast, finding a Nash equilibrium is PPAD-complete (i.e., *much* harder) [43–46]. The relation between the two sets is given by the following straightforward result:

Proposition 2.11 (Aumann, 1987)**.** *The set of correlated equilibria of a finite game is a nonempty convex polytope which contains the convex hull of the game's Nash equilibria.*

To the best of the author's knowledge, this inclusion seems to be the strongest universal statement relating these two notions of equilibrium.

COARSE CORRELATED EQUILIBRIUM. Going beyond correlated equilibria, the notion of a *coarse* correlated equilibrium replaces the pairwise comparison in the characterization (2.84) of correlated equilibria with a "coarse" averaging scheme. Tracing its origins to Moulin and Vial [104], we have:

Coarse correlated equilibrium

Definition 2.5. A correlated strategy π^* with law χ^* is a *coarse correlated equilibrium* (CCE) of the finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ if

$$u_i(\pi^*) \geq u_i(a'_i; \pi^*_i) \quad \text{for all } i \in \mathcal{N} \text{ and all } a'_i \in \mathcal{A}_i, \tag{CCE}$$

where, in a slight abuse of notation, a'_i denotes here the constant recommendation $\omega \mapsto a'_i$ for all $\omega \in \Omega$. More explicitly, χ^* is a CCE if

$$\sum_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} \chi^*_{a_i; a_{-i}} u_i(a_i; a_{-i}) \geq \sum_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} \chi^*_{a_i; a_{-i}} u_i(a'_i; a_{-i}) \tag{2.87}$$

for all $i \in \mathcal{N}$ and all $a'_i \in \mathcal{A}_i$. The set of coarse correlated equilibria of Γ will be denoted throughout as $\mathrm{CCE}(\Gamma)$.

Clearly, any correlated strategy satisfying (CE) also satisfies (CCE). Thus, in view of Proposition 2.10, we have the chain of inclusions

$$\mathrm{NE}(\Gamma) \subseteq \mathrm{NE}(\Delta(\Gamma)) \subseteq \mathrm{CE}(\Gamma) \subseteq \mathrm{CCE}(\Gamma) \tag{2.88}$$

where $\Delta(\Gamma)$ denotes the mixed extension of Γ. Importantly, these inclusions are usually proper: $\mathrm{NE}(\Gamma)$ is often empty while the cardinality of $\mathrm{NE}(\Delta(\Gamma))$ is generically odd [59];

by contrast, $\mathrm{CE}(\Gamma)$ and $\mathrm{CCE}(\Gamma)$ are both convex polytopes, but typically of different dimension.

CCE may contain dominated strategies

In particular, regarding the difference between Definitions 2.4 and 2.5 the key point is that the former considers *all* strategy modifications of π_i^* that are consistent with the information at hand, while the latter only considers *constant* strategy modifications that output the same pure action irrespective of the state of the world. Thus, given that coarse correlated equilibria are resilient only against a very narrow and specific class of strategy modifications, $\mathrm{CCE}(\Gamma)$ may contain correlated strategies that are not rationalizable. Indeed, Viossat and Zapechelnyuk [148] constructed an example of a (symmetric) 4×4 variant of Rock-Paper-Scissors which admits a coarse correlated equilibrium that assigns positive probability *only to strictly dominated strategies*; by contrasted, correlated equilibria cannot be supported on dominated strategies.

The above shows that the notion of a coarse correlated equilibrium is a fairly weak solution concept, barely deserving the appellation "equilibrium".[18] As we shall see in the sequel, coarse correlated equilibria are learnable by players that follow a no-regret policy [20, 35, 60];[19] however, this begs the question of what *exactly* is being learned. Much of the analysis to come focuses precisely on this question.

18 The set of coarse correlated equilibria is also called the *Hannan set* in reference to the original work of Hannan [57]. I find this terminology preferable, but the term coarse correlated equilibrium is fairly entrenched by now.

19 We should also note here that correlated equilibria are likewise learnable by players that follow a policy that leads to no *internal* regret [60, 61]. The notion of internal regret can be difficult to define in games with continuous action spaces, and the distinction between internal and external regret (as well as the link with calibration and universal consistency) lies beyond the scope of this manuscript, so we will not treat this issue here. For a relatively recent treatment, we refer the reader to Cesa-Bianchi et al. [36].

3

LEARNING IN GAMES: A CONTINUOUS-TIME SKELETON

In this and the following chapter, our aim is to examine the long-run behavior of online learning dynamics in a game-theoretic context. Specifically, we seek to address the following questions:

Does no-regret learning lead to rationally admissible states?
In particular, does it converge to a Nash equilibrium?

As we discussed towards the end of the previous chapter, the counterexample of Viossat and Zapechelnyuk [148] shows that there exist coarse correlated equilibria that are supported *only* on strictly dominated strategies. Therefore, since playing a coarse correlated equilibrium leads to no regret [35], the answer to both questions above is, in general, a resounding "no". Especially on the issue of convergence to Nash equilibrium, the impossibility result of Hart and Mas-Colell [62] shatters any hope of obtaining an unconditionally positive answer when the players' dynamics are *uncoupled* – i.e., the adjustment of a player's strategy does not depend explicitly on the payoff functions of the other players. All in all, as pointed out by Cesa-Bianchi and Lugosi [35, p. 205] specifying the precise interplay between no-regret learning and Nash equilibrium is a "considerably more difficult problem".

Learning in continuous time

In this chapter, our aim is to obtain partial positive answers to the above in a *continuous-time* framework, i.e., when the sequence of play unfolds over a continuous interval of time $t \in [0, \infty)$. The reason for this is both conceptual and technical: Conceptually, it allows us to connect our results to the extensive lite rature on game dynamics in biology and evolutionary game theory (EGT), thus providing an important link between learning and evolution in this context.[1] From a technical standpoint, the continuous-time setting often produces a clearer analytical picture which can subsequently serve as a scaffolding for the discrete-time analysis, while also demystifying some trade-offs that arise therein. We carry out the corresponding discrete-time analysis in Chapter 4.

3.1 LEARNING DYNAMICS

 # This section summarizes results from [80, 92, 97]

Regret in continuous time

Given our focus on no-regret learning, we begin by redefining the notion of regret in a continuous-time environment. Specifically, with notation as in the previous chapter, the regret incurred at time $T \geq 0$ by a policy $X_t \in \mathcal{X}$ against a stream of loss functions $\ell_t \colon \mathcal{X} \to \mathbb{R}$, $t \geq 0$, is now defined as

$$\mathrm{Reg}(T) = \max_{x \in \mathcal{X}} \int_0^T [\ell_t(X_t) - \ell_t(x)]\, dt \tag{3.1}$$

i.e., as the continuous-time analogue of Eq. (2.9).[2]

1 We do not attempt to survey this literature here. For an introduction, we refer the reader to the masterful accounts of Hofbauer and Sigmund [67], Weibull [151] and Sandholm [129, 130].

2 In the above, we tacitly assume that X_t and ℓ_t are both locally integrable in t so the integral in (3.1) is well-defined. This assumption plays no role in the sequel, so there is no sense in making it explicit.

Online mirror descent in continuous time

To minimize regret in this continuous-time framework, we will focus on the *continuous-time dual averaging* dynamics

$$\begin{aligned} \dot{Y}_t &= V_t \\ X_t &= Q(Y_t) \end{aligned} \tag{CDA}$$

i.e., the continuous-time analogue of the (lazy) mirror descent / dual averaging algorithm (DA) discussed in Section 2.2. In more detail, to account for the unilateral and multi-agent viewpoints that form the basis of our analysis, we will assume that the impulse process V_t is generated according to two different mechanisms as follows:

1. In the *unilateral setting*, a single optimizer is facing a stream of loss functions $\ell_t : \mathcal{X} \to \mathbb{R}$ and V_t is given by

$$V_t = -\nabla \ell_t(X_t). \tag{3.2}$$

2. In the *multi-agent setting*, we assume that the agents are involved in a continuous game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ and each player $i \in \mathcal{N}$ is facing a stream of payoff functions $u_i(\cdot, X_{-i,t})$ defined by the actions of i's opponents. In this way, we have

$$V_t = V(X_t) \tag{3.3}$$

where $V(x) = (V_i(x))_{i \in \mathcal{N}}$ is the players' individual payoff gradient profile (2.75).

In both cases, $Y_t \in \mathcal{Y} \equiv \mathcal{V}^*$ is an auxiliary dual variable that aggregates gradient signals as they arrive, and the mirror map $Q : \mathcal{Y} \to \mathcal{X}$ is induced by a distance-generating function on $\mathcal{X}$ as in (2.60). More explicitly, in the multi-agent case:

1. $X_t = (X_{i,t})_{i \in \mathcal{N}} \in \mathcal{X} \equiv \prod_i \mathcal{X}_i$ denotes the players' action profile at time $t \geq 0$.
2. $Y_{i,t} \in \mathcal{Y}_i \equiv \mathcal{V}_i^*$ is a player-specific gradient aggregation variable.
3. Each player is equipped with a mirror map $Q_i : \mathcal{Y}_i \to \mathcal{X}_i$ generated from a player-specific regularizer $h_i : \mathcal{X}_i \to \mathbb{R}$ as in (2.60).
4. The composite mirror map $Q : \mathcal{Y} \to \mathcal{X}$ is defined as

$$Q(y) = (Q_1(y_1), \dots, Q_N(y_N)) \quad \text{for all } y \in \mathcal{Y} \tag{3.4}$$

and is generated from the composite regularizer $h : \mathcal{X} \to \mathbb{R}$ with $h(x) = \sum_i h_i(x_i)$.

The above setup allows us to treat the unilateral and multi-agent cases with a unified language and notation. However, it is important to note a fundamental difference between the two: Specifically, in the multi-agent case, the dynamics (CDA) may be written more compactly as

$$\dot{Y}_t = V(Q(Y_t)) \tag{3.5}$$

Therefore, in the game-theoretic setup, (CDA) is an *autonomous* dynamical system evolving in the dual space $\mathcal{Y}$ of the ambient space $\mathcal{V} \equiv \prod_i \mathcal{V}_i$. By contrast, in the unilateral setting, the impulse process V_t may depend explicitly on t, so (CDA) is a non-autonomous system in that case.

Examples in finite games

The dynamics (CDA) will be the main focus of this chapter, so we discuss some basic examples below. For concreteness, we will state them for mixed extensions of finite games (cf. Examples 2.11 and 2.16), in which case the players' individual payoff gradient field is given by the concrete expression (2.76), viz.

$$V_{ia_i}(x) = \frac{\partial u_i}{\partial x_{ia_i}} = u_i(a_i; x_{-i}) \equiv u_{ia_i}(x). \tag{3.6}$$

We state two concrete examples of the induced dynamics below:

The replicator dynamics

Example 3.1. Going back to the entropic regularization framework of Example 2.10, we obtain the *exponential* (or *logit*) *learning dynamics*

$$\begin{aligned} \dot{Y}_{ia_i} &= u_{ia_i}(X) \\ X_{ia_i} &= \frac{\exp(Y_{ia_i})}{\sum_{a_i' \in \mathcal{A}_i} \exp(Y_{ia_i'})}. \end{aligned} \tag{XL}$$

In game theory, this process has been studied by Hofbauer et al. [69], Kwon and Mertikopoulos [80], Rustichini [126], Sorin [139], and many others. In particular, differentiating X_{ia_i} in (XL) with respect to time and substituting yields

$$\begin{aligned} \dot{X}_{ia_i} &= \frac{\dot{Y}_{ia_i} e^{Y_{ia_i}} \sum_{a_i' \in \mathcal{A}_i} e^{Y_{ia_i'}} - e^{Y_{ia_i}} \sum_{a_i' \in \mathcal{A}_i} \dot{Y}_{ia_i'} e^{Y_{ia_i'}}}{\left(\sum_{a_i' \in \mathcal{A}_i} e^{Y_{ia_i'}}\right)^2} \\ &= X_{ia_i} \left[\dot{Y}_{ia_i} - \sum_{a_i' \in \mathcal{A}_i} X_{ia_i'} \dot{Y}_{ia_i'} \right]. \end{aligned} \tag{3.7}$$

Hence, with $\dot{Y}_{ia_i} = u_{ia_i}(X)$, we obtain the *replicator dynamics* of Taylor and Jonker [143]:

$$\dot{X}_{ia_i} = X_{ia_i}[u_{ia_i}(X) - u_i(X)]. \tag{RD}$$

The replicator equation is the most widely studied of evolutionary game dynamics (by far), and its rationality properties have been the focus of an extensive literature in evolutionary game theory and population biology. For a survey, we refer the reader to Hofbauer and Sigmund [67], Maynard Smith and Price [88], Sandholm [129], Weibull [151], and references therein.

The projection dynamics

Example 3.2. In the Euclidean setup of Example 2.9, we get the projection-based learning process

$$\begin{aligned} \dot{Y}_{ia_i} &= u_{ia_i}(X) \\ X &= \Pi(Y) \end{aligned} \tag{PL}$$

where $\Pi(\cdot)$ denotes here the Euclidean projector on $\mathcal{X}$. Since Π is not smooth, we can no longer use the approach of Example 3.2 to derive the dynamics of the players' mixed strategies X_i. Instead, recall (or solve the defining convex program to show) that the closest point projection on $\mathcal{X}_i \equiv \Delta(\mathcal{A}_i)$ takes the simple form

$$[\Pi(y_i)]_{ia_i} = [y_{ia_i} + \mu_i]_+, \tag{3.8}$$

where $\mu_i \in \mathbb{R}$ is such that $\sum_{a \in \mathcal{A}_i} [y_{ia_i} + \mu_i]_+ = 1$. Therefore, if $_i$ is an open time interval over which X_i has constant support $\mathcal{A}_i' \subseteq \mathcal{A}_i$, a straightforward calculation detailed in [92] yields the so-called *projection dynamics:*

$$\dot{X}_{ia_i} = \begin{cases} u_{ia_i}(X) - |\operatorname{supp}(X_i)|^{-1} \sum_{a_i' \in \operatorname{supp}(X_i)} u_{ia_i'}(X) & \text{if } a_i \in \operatorname{supp}(X_i), \\ 0 & \text{if } a_i \notin \operatorname{supp}(X_i). \end{cases} \tag{PD}$$

These dynamics were introduced in game theory by Friedman [52] as a geometric model of the evolution of play in population games.[3] In contrast to (RD), orbits of (PL) that begin in $\operatorname{ri}(\mathcal{X})$ may attain a boundary face in finite time, then move to another boundary face or re-enter $\operatorname{ri}(\mathcal{X})$ (again in finite time), and so on (cf. Fig. 3.1). Thus,

3 [106] (see also [81] and [131]) introduce related projection-based dynamics for population games. The relations among the various projection dynamics are explored in a companion paper [87].

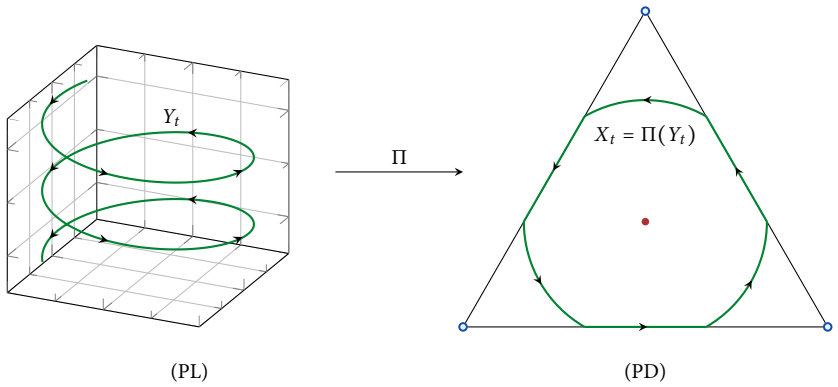


Figure 3.1: The primal-dual relation between (PL) and (PD).

although X_t may fail to be differentiable when it moves from (the relative interior of) one face of $\mathcal{X}$ to another, it satisfies (PD) for all times in between. This also shows that, while (PL) is an autonomous dynamical system, (PD) is not.

Forward invariance of the relative interior

The two examples above highlight an important dichotomy in the behavior of (CDA) in finite games: the replicator dynamics (RD) always remain in the (relative) interior of the game's strategy space, while the projection dynamics (PD) may enter and leave the boundary of $\mathcal{X}$ in perpetuity (so strategies that are extinct may reappear over time). As we argue below, this dichotomy has the same origin as the difference between the eager and lazy variants of (OMD).

Hessian–Riemannian considerations

To see this, assume that each player's distance-generating function h_i is twice differentiable on $\operatorname{ri}(\mathcal{X}_i)$, let

$$g_i(x_i) = \operatorname{Hess}(h_i(x_i)) \tag{3.9}$$

denote its Hessian matrix, and let

$$H_i(x_i) = g_i(x_i)^{-1} \tag{3.10}$$

denote its inverse. Assume further that H_i admits a continuous extension to each face of $\mathcal{X}_i$, and let

$$n_i(x_i) = H_i(x_i)\mathbf{1} \tag{3.11}$$

where $\mathbf{1} = (1, \dots, 1)$ is a column vector of ones of the appropriate dimension. Since h_i is strongly convex, g_i is positive-definite. As such, g_i can be seen as a Riemannian metric on $\operatorname{ri}(\mathcal{X}_i)$ and, under this metric, $n_i(x_i)$ is simply the unit normal to $\mathcal{X}_i$ at x_i. We then have the following explicit expression for the evolution of X_t in finite games:

The primal dynamics

Proposition 3.1 (Mertikopoulos and Sandholm, 2016)**.** *Let $X_t = Q(Y_t)$ be an orbit of* (CDA) *in $\mathcal{X}$, and let be an open interval over which the support of X_t remains constant. Then, for all $t \in$, X_t satisfies:*

$$\dot{X}_i = \left[H_i(X_i) - \frac{n_i(X_i)n_i(X_i)^\top}{\mathbf{1}^\top n_i(X_i)}\right] V_i(X). \tag{3.12}$$

In particular, every orbit $X_t = Q(Y_t)$ of (CDA) *in $\mathcal{X}$ is Lipschitz continuous and satisfies* (3.12) *on an open dense subset of $[0, \infty)$. Furthermore, if each h_i is steep in the sense of* (2.67)*, the system* (3.12) *is well-posed and X_t is an ordinary solution thereof.*

Remark 3.1. In game theory, the closest antecedent to (3.12) is the "escort replicator equation" of Harper [58]. From an optimization standpoint, (3.12) can also be seen as a game-theoretic analogue of the Hessian–Riemannian gradient system of Bolte

and Teboulle [21] and Alvarez et al. [4]. Similar considerations are present in the class of adaptive dynamics studied by Hofbauer and Sigmund [66] and, later, by Hopkins [70]; for a detailed discussion and a more extensive literature review on this topic, see Mertikopoulos and Sandholm [93].

3.2 NO-REGRET VS. CONVERGENCE

 # This section summarizes results from [80, 101]

3.2.1 *Regret minimization*

We now proceed to examine the no-regret properties of the dynamics (CDA) in the unilateral case. Our main result along these lines is as follows:

Constant regret in continuous time

Theorem 3.2 (Kwon and Mertikopoulos, 2017)**.** *Suppose that* (CDA) *is run against a locally integrable stream of convex loss functions $\ell_t : \mathcal{X} \to \mathbb{R}$, $t \geq 0$. Then, the optimizer's incurred regret is bounded as*

$$\operatorname{Reg}(T) \leq H \tag{3.13}$$

where $H = \max h - \min h$ denotes the depth of h over $\mathcal{X}$.

This "constant regret" result represents a significant improvement over the $\mathcal{O}(\sqrt{T})$ worst-case bound for (OMD)/(DA) in discrete time. For this reason, the continuous-time framework we consider here can be seen as particularly amenable to learning because it allows players to minimize their regret at the fastest possible rate – i.e., $\mathcal{O}(1)$. At the same time however, the $\Omega(\sqrt{T})$ minimax bound for regret minimization in discrete time, shows that there is an important gap between the continuous and discrete regimes.

Discrete vs. continuous time

This gap between continuous and discrete time was first observed in the context of the Hedge / EW algorithm by Sorin [139] who derived the associated discrete-time bound (2.56a) via a piecewise constant continuous-time approximation scheme. The approach of Sorin [139] was subsequently extended to general online convex optimization problems by Kwon and Mertikopoulos [80] who showed that the bound (2.51) can be decomposed as follows: the first term represents the regret incurred by the algorithm's continuous-time analogue, while the second term measures the "discretization error" when descending from continuous to discrete time. We discuss this in more detail in Section 4.1.

3.2.2 *Cycles, non-convergence, and Poincaré recurrence*

Empirical means and time-averages

In the context of finite games, an immediate corollary of Theorem 3.2 is that the empirical frequency of play under (CDA) converges to the game's set of CCE at a rate of $\mathcal{O}(1/t)$. Specifically, given a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ and a pure action profile $a = (a_1, \dots, a_N) \in \mathcal{A}$, let

$$Z_{a,t} = \frac{1}{t} \int_0^t \prod_{i \in \mathcal{N}} X_{i a_i, s} \, ds \tag{3.14}$$

denote the mean ("empirical") frequency of a under the policy $X_t \in \mathcal{X}$. Then, if X_t is generated by (CDA), the constant regret bound (3.13) implies that Z_t converges to $\operatorname{CCE}(\Gamma)$ at a rate of $1/t$ [20]. On the other hand, as we pointed out before, $\operatorname{CCE}(\Gamma)$ may contain correlated strategies that are supported only on strictly dominated strategies, so this convergence result is fairly weak.

In particular, two natural questions that arise are:

1. Is the long-run behavior of X_t captured by that of its time-average

$$X_t = \frac{1}{t}\int_0^t X_s\, ds \tag{3.15}$$

or that of its correlated empirical mean Z_t?

2. Is the limit behavior of X_t rationally admissible?

To take a closer look at these questions, a key notion will be that of *Poincaré recurrence*. Heuristically, a dynamical system is said to be recurrent if, after a sufficiently long (but *finite*) time, almost every state returns arbitrarily close to the system's initial state.[4] More formally, given a dynamical system on $\mathcal{X}$ that is defined by means of a *semiflow* $\Phi\colon \mathcal{X}\times[0,\infty)\to\mathcal{X}$, we have:[5]

Poincaré recurrence

Definition 3.1. A point $x\in\mathcal{X}$ is said to be *recurrent* under Φ if, for every neighborhood U of x in $\mathcal{X}$, there exists an increasing sequence of times $t_k\uparrow\infty$ such that $\Phi(x,t_k)\in U$ for all k. In addition, the flow Φ is called itself *Poincaré recurrent* if, for every measurable subset A of $\mathcal{X}$, the set of recurrent points in A has full measure.

An immediate consequence of Definition 3.1 is that, if a point is recurrent, there exists an increasing sequence of times $t_k\uparrow\infty$ such that $\Phi(x,t_k)\to x$. On that account, recurrence can be seen as the flip side of convergence: under the latter, (almost) every initial state of the dynamics would eventually reach a well-defined end-state; instead, under the former, the system's orbits fill the entire state space and return arbitarily close to their starting points infinitely often (so there is no possibility of convergence beyond trivial cases).

As we show below, no-regret learning may exhibit recurrent behavior, even in simple, 2-player games:

Non-convergence in zero-sum games

Theorem 3.3 (Mertikopoulos et al., 2018)**.** *Let Γ be a finite 2-player zero-sum game admitting an interior Nash equilibrium. Then, almost every solution trajectory of* (CDA) *is Poincaré recurrent; specifically, for almost every initial condition $X_0=Q(Y_0)\in\mathcal{X}$, there exists an increasing sequence of times $t_k\uparrow\infty$ such that $X_{t_k}\to X_0$.*

A key element in the proof of Theorem 3.3 is that, in 2-player zero-sum games, the dynamics (CDA) admit a constant of motion. This is given by the so-called Fenchel coupling,[6] defined here as

$$F(x,y) = h(x) + h^*(y) - \langle y, x\rangle \quad \text{for all } x\in\mathcal{X},\ y\in\mathcal{Y}. \tag{3.16}$$

Specifically, if x^* is an interior equilibrium of Γ, we have $F(x^*,Y_t)=F(x^*,Y_0)$ for all $t\geq 0$. In fact, the invariance of (3.16) under (CDA) induces a foliation of $\mathcal{Y}$, with each individual orbit of (CDA) living on a "leaf" of the foliation (a level set of F). Fig. 3.2 provides a schematic illustration of this foliation/cycling structure in a game of Matching Pennies.

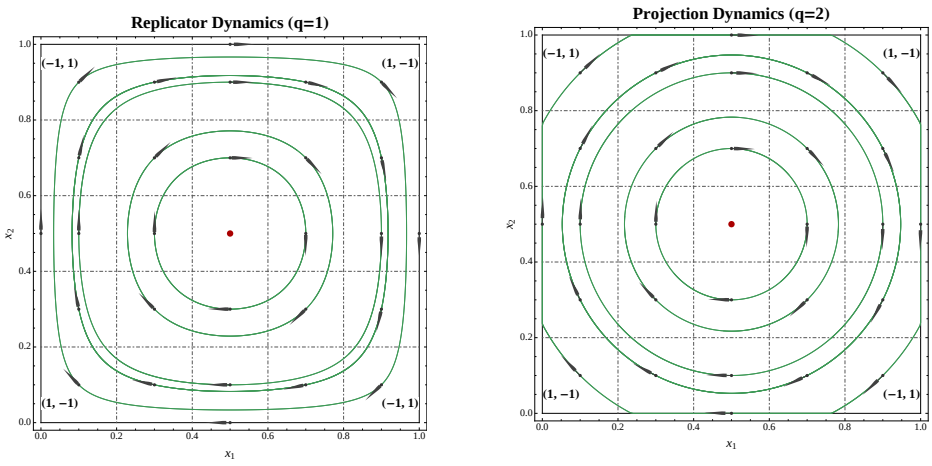
Convergence of time-averages

We close this section by noting that the behavior of the time-averaged orbits $\bar{X}_t$ of (CDA) is considerably different. In fact, as was shown by Hofbauer et al. [69] and Mertikopoulos and Sandholm [92], the time-avaraged orbits of (CDA) have the same limit as the best-response dynamics of Gilboa and Matsui [55]. Consequently, in zero-sum games with an interior equilibrium, $\bar{X}_t$ always converges to Nash equilibrium,

4 Here, "almost" means that the set of such states has full Lebesgue measure.

5 Recall here that a continuous map $\Phi\colon\mathcal{X}\times[0,\infty)\to\mathcal{X}$ is a *semiflow* if $\Phi(x,0)=x$ and $\Phi(x,t+s)=\Phi(\Phi(x,t),s)$ for all $x\in\mathcal{X}$ and all $s,t\geq 0$.

6 The terminology "Fenchel coupling" is taken from [97]; the link between the Fenchel coupling and the Bregman divergence is also discussed therein. Specifically, we have $F(x,y)=D(x,Q(y))$ whenever $Q(y)\in\operatorname{ri}(\mathcal{X})$, but not necessarily otherwise.



(a) Replicator dynamics (XL) / (RD) **(b)** Projection dynamics (PL) / (PD)

Figure 3.2: Cycles and recurrence of no-regret learning in zero-sum games.

even though the actual trajectories of play may remain at constant distance from said equilibrium. This disparity between the actual sequence of play and its time-average will appear several times in the sequel and it should be taken as an important cautionary tale for the "convergence" of no-regret learning in games.

3.3 CONVERGENCE TO EQUILIBRIUM AND RATIONALIZABILITY

 # This section summarizes results from [92, 93, 97]

We now proceed in the opposite direction, i.e., establishing positive results for the rationality properties of (CDA). To connect our discussion with that of the previous section, we first present a series of results for finite games and we then move on to continuous games towards the end of this section.

3.3.1 *Positive results in finite games*

Dominated strategies

We begin our rationality analysis with the elimination of dominated strategies. Formally, given a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$, we say that $a_i \in \mathcal{A}_i$ is dominated by $a_i' \in \mathcal{A}_i$ (and we write $a_i \prec a_i'$) if

$$u_{ia_i}(x) \equiv u_i(a_i; x_{-i}) < u_i(a_i'; x_{-i}) \quad \text{for all } x_{-i} \in \mathcal{X}_{-i} \equiv \textstyle\prod_{j\neq i} \mathcal{X}_j. \tag{3.17}$$

If (3.17) is strict for only some (but not all) $x \in \mathcal{X}$, we will say that a_i is *weakly dominated* by a_i' and we will write $a_i \preccurlyeq a_i'$. Conversely, we will say that $x = (a_1, \dots, a_N) \in \mathcal{A}$ is *undominated* if no component of a is (strictly) dominated. Of course, if dominated strategies strategies are removed from Γ, other strategies may become dominated in the resulting restriction of Γ, leading to the notion of an *iteratively dominated strategy*. A strategy which survives all rounds of elimination is then called *iteratively undominated*. Finally, given a trajectory of play $X_t \in \mathcal{X}$, $t \geq 0$, we say that $a_i \in \mathcal{A}_i$ *becomes extinct along* X_t if $X_{ia_i,t} \to 0$ as $t \to \infty$.

Extending the classic elimination results of Akin [3], Nachbar [105] and Samuelson and Zhang [127] for the replicator dynamics, we show below that only iteratively undominated strategies survive under any no-regret learning scheme of the general form (CDA):

Extinction of dominated strategies

Theorem 3.4 (Mertikopoulos and Sandholm, 2016). *Let $X_t = Q(Y_t)$ be an orbit of*

(CDA). *If $a_i \in \mathcal{A}_i$ is dominated (even iteratively), then it becomes extinct along X_t. More explicitly, suppose that each player's regularizer is of the form*

$$h_i(x_i) = \sum_{a_i \in \mathcal{A}_i} \theta_i(x_{ia_i}) \tag{3.18}$$

for some continuous, strictly convex function $\theta_i : [0,1] \to \mathbb{R}$ that is smooth on $(0,1]$. If $a_i \prec a_i'$, then

$$X_{ia_i,t} \leq \phi_i(c_i - \delta_i t), \tag{3.19}$$

where

$$\delta_i = \min\{u_{ia_i'}(x) - u_{ia_i}(x) : x \in \mathcal{X}\} \tag{3.20}$$

is the minimum payoff difference between a_i and a_i', c_i is a constant that only depends on the dynamics' initial conditions, and the rate function ϕ_i is given by

$$\phi_i(z) = \begin{cases} 0 & \text{if } z \leq \theta_i'(0^+), \\ 1 & \text{if } z \geq \theta_i'(1^-), \\ (\theta_i')^{-1}(z) & \text{otherwise,} \end{cases} \tag{3.21}$$

where $(\theta_i')^{-1}$ is the inverse function of θ_i'.

Elimination in (RD) *and* (PD)

Corollary 3.5. *Under the replicator dynamics* (RD)*, dominated strategies become extinct at a geometric $\mathcal{O}(\exp(-\delta t))$ rate. By contrast, under the projection dynamics* (PD)*, dominated strategis become extinct in finite time.*

Remark 3.2. The significance of this result is that it shows that no-regret learning dynamics (CDA) are not vulnerable to the counterexample of Viossat and Zapechelnyuk [148]. Even though CCE(Γ) might contain correlated strategies that are supported only on strictly dominated strategies, Theorem 3.4 shows that no-regret learning based on (OMD) *does* avoid this part of CCE(Γ) – in continuous time at least.

Remark 3.3. For posterity, we should also mention here that "rates" in a continuous-time setting are not necessarily meaningful, because the time parameter can be reparametrized arbitrarily. Nevertheless, Theorem 3.4 does admit a matching result in discrete time, which we state in Chapter 4.

Stability and convergence

We now turn to the equilibrium convergence properties of (CDA). Heuristically, these can be summarized as follows:

1. Nash equilibria are stationary under (CDA).
2. If an interior solution orbit converges, its limit is a Nash equilibrium.
3. If a point is stable under (CDA), then it is a Nash equilibrium.
4. Strict Nash equilibria are stable and attracting under (CDA).

Of course, since (CDA) does not evolve directly on $\mathcal{X}$ (and the associated primal dynamics may fail to be well-posed if a player's regularizers is not steep), the standard notions of dynamical stability and stationarity must be modified accordingly. In particular, these notions continue to apply in the dual space $\mathcal{V}^*$; however, since the mapping $Q : \mathcal{Y} \to \mathcal{X}$ is neither injective nor surjective, this approach would not suffice to define stationarity and stability on $\mathcal{X}$. We address this issue via the following definition:

Definition 3.2. Fix $x^* \in \mathcal{X}$ and let $X_t = Q(Y_t)$ be an orbit of (CDA). We say that:

1. x^* is *stationary* under (CDA) if $X_t = x^*$ for all $t \geq 0$ whenever $X_0 = x^*$.
2. x^* is *Lyapunov stable* under (CDA) if, for every neighborhood U of x^*, there exists a neighborhood U' of x^* such that $X_t \in U$ for all $t \geq 0$ whenever $X_0 \in U'$.

3. x^* is *attracting* under (CDA) if it admits a neighborhood U such that $X_t \to x^*$ as $t \to \infty$ whenever $X_0 \in U$.
4. x^* is *asymptotically stable* under (CDA) if it is Lyapunov stable and attracting.

Stationarity in the primal vs. stationarity in the dual

On the issue of stationarity, there are some subtle points that arise. First, note that x^* is implicitly required to belong to $\mathcal{X}^\circ$ (since $x^* = X_0 = Q(Y_0)$); however, no such assumption is made for stable and/or attracting states. From a dynamical standpoint, the reason for this distinction is that stationary points should be (constant) trajectories of the dynamical system under study, whereas Lyapunov stable and attracting states only need to be *approachable* by orbits. Clearly, since $\operatorname{ri}(\mathcal{X}) \subseteq \mathcal{X}^\circ$, any point in $\mathcal{X}$ can be a candidate for (asymptotic) stability under (CDA). However, boundary points might not be suitable candidates for stationarity, so stability does *not* imply stationarity (as would be the case for a bona fide dynamical system defined on $\mathcal{X}$).

With this definition in hand, we have the following basic result:

Equilibrium convergence and stability properties

Theorem 3.6 (Mertikopoulos and Sandholm, 2016)**.** *Fix $x^* \in \mathcal{X}$ and let $X_t = Q(Y_t)$ be an orbit of* (CDA)*. Then:*

1. *If $x^* \in \mathcal{X}$ is stationary under* (CDA)*, it is a Nash equilibrium; conversely, if x^* is a Nash equilibrium and $x^* \in \operatorname{im} Q$, then x^* is stationary under* (CDA)*.*
2. *If $X_t \to x^*$ as $t \to \infty$, then x^* is a Nash equilibrium.*
3. *If $x^* \in \mathcal{X}$ is Lyapunov stable under* (CDA)*, then x^* is a Nash equilibrium.*
4. *If x^* is a strict Nash equilibrium, it is also asymptotically stable under* (CDA)*.*

To the best of the author's knowledge, this is the strongest universal convergence result that can be obtained for the no-regret dynamics (CDA) in finite games – i.e., without further assumptions on the structure of the game being played. For an in-depth discussion, we refer the reader to Mertikopoulos and Sandholm [92, 93].

3.3.2 *Positive results in concave games*

With these first results in place for finite games, we turn to the long-run behavior of the no-regret dynamics (CDA) in *concave* games (i.e., continuous games with individually concave payoff functions). Since the notion of strategic dominance does not really apply beyond finite games, we will focus on the dynamics' equilibrium convergence properties.

Blanket assumptions

To state our result, we will make the following general assumptions:

Assumption 3.1 (Concavity)**.** The underlying game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ is concave.

Assumption 3.2 (Lipschitz smoothness)**.** The individual payoff gradient field $V\colon \mathcal{X} \to \mathbb{R}$ is Lipschitz continuous, i.e.,

$$\|V(x') - V(x)\|_* \leq \beta \|x' - x\| \tag{3.22}$$

for some $\beta > 0$ and all $x, x' \in \mathcal{X}$.

Assumption 3.3 (Fenchel reciprocity)**.** For all $x \in \mathcal{X}$ and every sequence $y_i \in \mathcal{Y}$, $k = 1, 2, \dots$, the Fenchel coupling (3.16) satisfies the *reciprocity condition*

$$F(x, y_i) \to 0 \quad \text{whenever} \quad Q(y_i) \to x. \tag{3.23}$$

Remark. The term "reciprocity" is due to [97] and expresses the following topological equivalence: By the strong convexity assumption for h, it follows that

$$F(x, y) \geq \frac{K}{2} \|Q(y) - x\|^2 \quad \text{for all } x \in \mathcal{X}, y \in \mathcal{Y}. \tag{3.24}$$

As a result, if $F(x, y_k) \to 0$ as $k \to \infty$, we will also have $x_k \equiv Q(y_k) \to x$. Assumption 3.3 posits the converse to this statement, hence the name "reciprocity". Similar conditions for the Bregman divergence have been considered by Alvarez et al. [4], Kiwiel [77] and many others. It is easy to verify that the regularizers considered so far all satisfy this condition.

With all this in hand, we have the following result:

Convergence in concave games

Theorem 3.7 (Mertikopoulos and Zhou, 2019)**.** *Suppose that Assumptions 3.1–3.3 hold, and let $X_t = Q(Y_t)$ be an orbit of* (CDA)*. Then:*

1. *If $X_t \to x^* \in \mathcal{X}$ as $t \to \infty$, then x^* is a Nash equilibrium of $\mathcal{G}$.*
2. *If, in addition, $\mathcal{G}$ is strictly monotone, X_t converges to a Nash equilibrium.*

In contrast to Theorem 3.14, the convergence guarantee of Theorem 3.7 is global, i.e., it is valid for any initial condition in (CDA). The price for this global convergence is the monotonicity requirement for V; this condition can actually be relaxed, but not without positing some modicum of global structure. In the absence of a global monotonicity requirement, it is still possible to derive local convergence results for (CDA) in the spirit of Theorem 3.14; for a detailed analysis, we refer the reader to [97].

3.4 LEARNING IN THE PRESENCE OF NOISE

 # This section summarizes results from [25, 90, 94–96]

Throughout the previous section, we implicitly assumed that (CDA) was run with access to perfect gradient information. However, for the same reasons that perfect oracle feedback is often hard to come by, this assumption is often violated in practical applications of game-theoretic learning: for instance, in telecommunication networks and traffic engineering, signal strength and latency measurements are constantly subject to stochastic fluctuations which introduce noise to the input of any learning algorithm. On that account, our aim in this section will be to examine the robustness of this analysis in the presence of stochastic perturbations to these measurements.

Dual averaging in the presence of noise

In our continuous-time framework, the most straightforward way to account for such perturbations is by introducing a Wiener process in Eq. (CDA). In so doing, we obtain the *stochastic dual averaging* dynamics

$$\begin{aligned} dY_t &= V_t\,dt + \sigma(X_t, t)\,dW_t \\ X_i &= Q(\eta_t Y_t) \end{aligned} \tag{SDA}$$

where the drift process V_t is defined as in the previous section,[7] W_t is a Wiener process in a Euclidean space $\mathcal{W} \cong \mathbb{R}^m$ and the (possibly state-dependent) diffusion matrix $\sigma\colon \mathcal{X} \to \operatorname{Hom}(\mathcal{V}, \mathcal{W}) \cong \mathbb{R}^{n\times m}$ measures the strength of the noise process (and is assumed to be Lipschitz continuous throughout).

The role of η_t

In addition to the Wiener process W_t, an important difference between (SDA) and (CDA) is the variable "learning rate" $\eta_t > 0$. The role of this parameter is discussed in detail below and, throughout the sequel, we make the following standing assumption

$$\eta_t \text{ is nonincreasing, } C^1\text{-smooth in } t, \text{ and } \lim_{t\to\infty} \eta_t = \infty. \tag{3.25}$$

Heuristically, the role of the learning parameter η_t in (SDA) is to temper the growth of the gradient aggregation variable Y_t so as to allow a better exploration of the problem's state space. In that regard, η_t should be contrasted to the vanishing step-size rules that are

7 That is, $V_t = -\nabla \ell_t(X_t)$ in the unilateral setting and $V_t = V(X_t)$ in the multi-agent setting.

used in the theory of stochastic approximation – see e.g., Benaïm [14], Borkar [22], Ljung [86], Robbins and Monro [120], and references therein. The difference between the two is that, in stochastic approximation, a variable step-size means that new information enters the algorithm with a decreasing weight; by contrast, in our context, all information would be weighed evenly, but the *aggregate* signal would be decreased by η_t in order to avoid extreme reponses to a given stimulus. This "post-moderation" of gradient signals is not needed in the deterministic setting of (CDA) but, as we shall see below, it plays a crucial role in the case of (SDA).

Stochastic replicator dynamics

Example 3.3. Going back to the entropic regularization framework of Example 2.10, (SDA) gives the stochastic exponential learning dynamics

$$\begin{aligned} dY_{ia_i} &= u_{ia_i}(X)\,dt + \sigma_{ia_i}\,dW_{ia_i} \\ X_{ia_i} &= \frac{\exp(Y_{ia_i})}{\sum_{a_i'\in\mathcal{A}_i}\exp(Y_{ia_i'})} \end{aligned} \tag{SXL}$$

where, for simplicity, we assume that W is a standard Wiener process in $\mathcal{Y}$, σ is diagonal, and we have suppressed the time index t. A straightforward application of Itô's lemma (see [25] for the details) then leads to the stochastic replicator dynamics

$$\begin{aligned} dX_{ia_i} &= \eta X_{ia_i}\left[u_{ia_i}(X) - \sum_{a_i'\in\mathcal{A}_i} X_{ia_i'}\,u_{ia_i'}(X)\right]dt \\ &\quad + \eta X_{ia_i}\left[\sigma_{ia_i}\,dW_{ia_i} - \sum_{a_i'\in\mathcal{A}_i}\sigma_{ia_i'}X_{ia_i'}\,dW_{ia_i'}\right] \\ &\quad + \frac{\dot\eta}{\eta}X_{ia_i}\left[\log X_{ia_i} - \sum_{a_i'\in\mathcal{A}_i} X_{ia_i'}\log X_{ia_i'}\right]dt \\ &\quad + \frac{\eta^2}{2}X_{ia_i}\left[\sigma_{ia_i}^2(1-2X_{ia_i}) - \sum_{a_i'\in\mathcal{A}_i}\sigma_{ia_i'}^2 X_{ia_i'}(1-2X_{ia_i'})\right]dt. \end{aligned} \tag{SRD}$$

Replicator dynamics with aggregate shocks

For constant η, (SRD) is simply the stochastic replicator dynamics of exponential learning first introduced in [90]. As such, (SRD) should be contrasted to the evolutionary *replicator dynamics with aggregate shocks* of [53]:

$$\begin{aligned} dX_{ia_i} &= X_{ia_i}\left[u_{ia_i}(X) - \sum_{a_i'\in\mathcal{A}_i} X_{ia_i'}u_{ia_i'}(X)\right]dt \\ &\quad + X_{ia_i}\left[\sigma_{ia_i}dW_{ia_i} - \sum_{a_i'\in\mathcal{A}_i}\sigma_{ia_i'}X_{ia_i'}\,dW_{ia_i'}\right] \\ &\quad - X_{ia_i}\left[\sigma_{ia_i}^2X_{ia_i} - \sum_{a_i'\in\mathcal{A}_i}\sigma_{ia_i'}^2X_{ia_i'}^2\right]dt, \end{aligned} \tag{ASRD}$$

where X_{ia_i} denotes the population share of the a_i-th genotype of species k in a multi-species environment, u_{ia_i} represents its reproductive fitness, and the noise coefficients σ_{ia_i} measure the impact of random weather-like effects on population evolution.[8] Besides the absence of the learning rate η, the fundamental difference between (SRD) and (ASRD) is the Itô correction in the last line of (SRD)/(ASRD). This difference is due to the different propagation of stochastic perturbations under (SDA) and it leads to drastic differences in the long-run behavior of (SRD) and (ASRD). For a detailed discussion of

8 For a comprehensive account of the literature surrounding (ASRD), see Hofbauer and Imhof [64], Mertikopoulos and Viossat [96], and references therein.

the differences between learning and evolution in this context, we refer the reader to Mertikopoulos and Viossat [96].

3.4.1 *Single-agent considerations*

With these preliminaries in hand proceed to establish the basic regret guarantees of (SDA) in a unilateral setting:

Regret incurred by (SDA)

Theorem 3.8 (Mertikopoulos and Staudigl, 2018)**.** *Suppose that* (SDA) *is run against a locally integrable stream of convex loss functions $\ell_t\colon \mathcal{X} \to \mathbb{R}$ with a variable parameter η_t satisfying* (3.25)*. Then, with probability* 1*, the incurred regret is bounded as*

$$\mathrm{Reg}(T) \leq \frac{H}{\eta_T} + \frac{\sigma_*^2}{2K}\int_0^T \eta_t\,dt + \mathcal{O}(\sqrt{T\log\log T}) \tag{3.26}$$

where $\sigma_*^2 = \max_x \|\sigma(x)\|_F^2$ and $H = \max h - \min h$ denotes the depth of h over $\mathcal{X}$. In particular, if (SDA) *is run with a learning rate of the form $\eta_t \propto t^{-p}$ for some $p \in (0,1)$, we have*

$$\mathrm{Reg}(T) = \begin{cases} \mathcal{O}(T^{1-p}) & \text{if } 0 < p < 1/2, \\ \mathcal{O}(\sqrt{T\log\log T}) & \text{if } p = 1/2, \\ \mathcal{O}(T^p) & \text{if } 1/2 < p < 1. \end{cases} \tag{3.27}$$

Remark. We should note here that Theorem 3.8 has been stated for a typical realization of W_t. The $\mathcal{O}(\sqrt{T\log\log T})$ term is actually a martingale term which vanishes in expectation, so the dynamics' also enjoy the bound

$$\overline{\mathrm{Reg}}(T) \leq \frac{H}{\eta_T} + \frac{\sigma_*^2}{2K}\int_0^T \eta_t\,dt \tag{3.28}$$

which allows us to get rid of the iterated logarithm factor $\sqrt{\log\log T}$.

The gap between (CDA) *and* (SDA)

Compared to the deterministic $\mathcal{O}(1)$ bound of Theorem 3.2, the bound (3.26) indicates a decrease in performance by a factor of $\tilde{\Theta}(\sqrt{T})$. The reason for this is the Itô correction term $\sigma_*^2/(2K)\int_0^T \eta_t\,dt$ in (3.26): balancing this second-order error against the noise-free term H/η_T imposes a $\Theta(1/\sqrt{t})$ schedule for η_t (otherwise, one term would be asymptotically slower than the other. In this regard, (3.26) is reminiscent of the $\mathcal{O}(1/\sqrt{T})$ bounds for (DA) derived in Chapter 2. In particular, as discussed earlier in this section, the increase in regret from $\mathcal{O}(1)$ to $\mathcal{O}(\sqrt{T})$ in discrete time stems from the discretization of the continuous-time dynamics which introduces a second-order Taylor term which makes constant regret unattainable. In the case of (SDA), there is no discretization gap, but the second-order correction in Itô's lemma ends up playing a similar role.

Ergodic convergence in static problems

We close this section with an examination of the properties of (SDA) in the static optimization framework of Example 2.1, i.e., when the optimizer is facing the same loss function $\ell_t = f$ for all $t \geq 0$. In this case, we are of course more interested about the convergence of (SDA) to an optimizer of f; the following corollary is an immediate consequence of Theorem 3.8:

Corollary 3.9 (Mertikopoulos and Staudigl, 2018)**.** *Suppose that* (SDA) *is run against the static optimization problem* (Opt) *with variable learning rate $\eta_t = \sqrt{HK}\sigma_*^2 \min\{1, \sqrt{t}\}$. Then, the time-averaged process $\bar{X}_t = (1/t)\int_0^t X_s\,ds$ enjoys the bound*

$$\mathbb{E}[f(\bar{X}_t)] \leq \min f + 2\sqrt{\frac{H\sigma_*^2}{Kt}}. \tag{3.29}$$

Because of the inherent stochasticity in (SDA), obtaining almost sure convergence results for the actual trajectories X_t is, in general, not possible (we examine this issue in

more detail towards the end of this section). As such, our goal in what follows will be to analyze the long-run concentration properties of (SDA) and to determine the domain that X_t occupies with high probability in the long run. For reasons that will become clear shortly, we focus on strongly convex problems with an interior solution $x^* \in \mathrm{ri}(\mathcal{X})$ and we will assume for simplicity that $\sigma \equiv \sigma_* I$ for some constant $\sigma_* > 0$.[9] Our first result in this context is as follows:

Hitting time statistics

Proposition 3.10 (Mertikopoulos and Sandholm, 2018)**.** *Let f be an α-strongly convex function with an interior minimizer $x^* \in \mathrm{ri}(\mathcal{X})$. If $X_t = Q(\eta Y_t)$ is an orbit of* (SDA) *initialized at $Y_0 = 0$, we have*

$$\mathbb{E}\left[\frac{1}{t}\int_0^t \|X_s - x^*\|^2\, ds\right] \leq \frac{2H}{\eta\alpha t} + \frac{\eta\sigma_*^2}{\alpha K}. \tag{3.30}$$

Moreover, if (SDA) *is run with $\eta = \alpha K\delta^2/(2\sigma_*^2)$ and $\tau_\delta = \inf\{t > 0 : \|X_t - x^*\| \leq \delta\}$ denotes the first time at which X_t gets within $\delta > 0$ of x^*, we have the hitting time estimate*

$$\mathbb{E}[\tau_\delta] \leq \frac{8H\sigma^2}{\alpha^2 K\delta^4}. \tag{3.31}$$

Remark 3.4. For a value-based analogue of (3.30) when h is steep, see Raginsky and Bouvrie [118, Prop. 4].

Proposition 3.10 provides a basic estimate of the long-run concentration of X_t around x^*, and also highlights the role of α and σ. Specifically, (3.31) shows that X_t hits a δ-neighborhood of x^* in time which is on average $\mathcal{O}(1/\delta^4)$. What's more, the multiplicative constant in this bound increases with the noise level in (SDA) and decreases with the sharpness of the minimum point x^* (as quantified by the strong convexity constant α of f). To obtain finer information regarding the concentration of X_t around x^*, we need to consider its *occupation measure*:

The occupation measure of X_t

Definition 3.3. The *occupation measure* of X_t at time $t \geq 0$ is given by

$$\mu_t(A) = \frac{1}{t}\int_0^t \mathbb{1}(X_s \in A)\, ds \quad \text{for every Borel } A \subseteq \mathcal{X}. \tag{3.32}$$

In words, $\mu_t(A)$ is the fraction of time that X spends in A up to time t. As such, the asymptotic concentration of X around x^* can be estimated by the quantity $\mu_t(\mathbb{B}_\delta)$, where

$$\mathbb{B}_\delta \equiv \mathbb{B}_\delta(x^*) = \{x \in \mathcal{X} : \|x - x^*\| \leq \delta\} \tag{3.33}$$

is the intersection of a δ-ball centered at x^* with $\mathcal{X}$. We then have the following concentration result (for a numerical illustration, see Fig. 3.3):

Long-run concentration around interior solutions

Theorem 3.11 (Mertikopoulos and Staudigl, 2018)**.** *Suppose that* (SDA) *is run against an α-strongly convex function admitting an interior minimizer $x^* \in \mathrm{ri}(\mathcal{X})$. Moreover, fix some $\delta > 0$ and suppose that the infinitesimal covariance matrix $\Sigma = \sigma\sigma^\top$ of* (SDA) *is time-homogeneous and uniformly positive-definite. Then, with probability* 1*, we have:*

$$\mu_t(\mathbb{B}_\delta) \gtrsim 1 - \frac{\eta\sigma^2}{\alpha K\delta^2} \quad \textit{for sufficiently large } t. \tag{3.34}$$

provided that $\eta < \alpha K\delta^2/\sigma^2$.

Corollary 3.12. *Fix some tolerance $\varepsilon > 0$. If* (SDA) *is run with assumptions as above and $\eta \leq \varepsilon\alpha K\delta^2/\sigma^2$, we have $\mu_t(\mathbb{B}_\delta) \geq 1 - \varepsilon$ for all sufficiently large t (a.s.).*

9 In fact, it suffices to assume that $\sup_{x,t}\|\sigma(x,t)\sigma(x,t)^\top\| \leq \sigma_*^2$; to streamline our discussion, we focus on the simplest case.

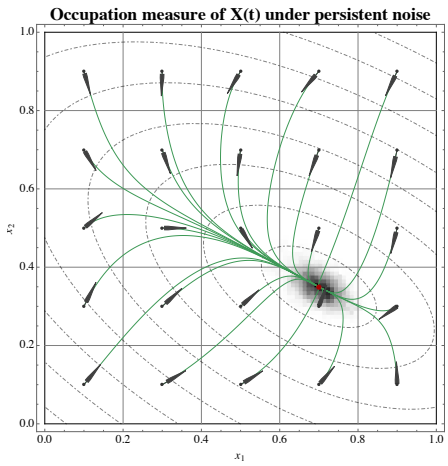


Figure 3.3: Numerical illustration of Theorem 3.11 regarding the long-run occupation measure of X_t under (SDA). The dashed contours represent the level sets of f over $\mathcal{X} = [0,1]^2$, and the flowlines indicate the flow of (CDA). The shades of gray correspond to higher probabilities of observing X_t in a given region (darker indicates higher probability).

Concentration and invariant measures

In a nutshell, Theorem 3.11 states that the concentration of X_t around x^* may be arbitrarily sharp if η is taken small enough. Indeed, for $\eta < \alpha K \delta^2/\sigma^2$, Proposition 3.10 shows that $\mathbb{B}_\delta$ is *recurrent*, i.e. $\mathbb{P}(X_t \in \mathbb{B}_\delta \text{ for some } t \geq 0) = 1$. In fact, it can be shown that the stated assumptions guarantee the existence of a unique invariant distribution ν for the dual process Y_t. The pushforward of ν to $\mathcal{X}$ is precisely the limit of the occupation measures μ_t of X_t as $t \to \infty$, so (3.34) follows by using the mean square bound (3.30) to estimate ν.

It is also worth noting that the bound (3.34) only depends on the mirror map Q via its inverse Lipschitz constant K (that is, the strong convexity constant of h). Eq. (3.34) suggests that K should be taken as large as possible so as to have $\mu_t(\mathbb{B}_\delta) \approx 1$. However, in so doing, the process X_t may initially spend a larger amount of time near the prox-center $x_c \equiv \arg\min h$ of $\mathcal{X}$. This is an important trade-off between the sharpness of the asymptotic concentration of X_t near x^* and the time it takes to attain this asymptotic regime.

3.4.2 *Multi-agent considerations*

We now turn to the study of (SDA) in a game-theoretic context. As in Section 3.3, we first present our analysis for finite games and then discuss continuous games towards the end of this section.

Extinction of dominated strategies

RESULTS FOR FINITE GAMES. We begin by examining the extinction of dominated strategies under (SDA). Despite the strong elimination properties of the deterministic dynamics (CDA), the extinction of dominated strategies can be a fairly subtle issue in the presence of noise and uncertainty. In the replicator dynamics with aggregate shocks (ASRD), Cabrales [33], Imhof [71] and Hofbauer and Imhof [64] provided a set of sufficient conditions on the intensity of the noise that guarantee the elimination of dominated strategies; however, if these conditions are not met, dominated strategies may – and, in fact, do – survive in the long run [96].

On the other hand, Mertikopoulos and Moustakas [90] showed that the noisy replicator dynamics (SRD) induced by (SDA) eliminate all strategies that are not iteratively undominated, irrespective of the noise level. As we show below, this unconditional elimination result extends to the entire class of no-regret dynamics covered by (SDA):

Theorem 3.13 (Bravo and Mertikopoulos, 2017)**.** *Let $X_t = Q(Y_t)$ be a solution orbit of* (SDA)*. If $a_i \in \mathcal{A}_i$ is dominated* (*even iteratively*)*, it becomes extinct along X_t with probability* 1*.*

Remark 3.5. Theorem 3.13 shows that (SDA) eliminated dominated strategies but it does not give any information on the rate of extinction – or how probable it is to observe a dominated strategy above a given level at some $t \geq 0$. This level of detail lies beyond the scope of this manuscript; for a detailed discussion and a precise result on these issues, we refer the interested reader to Bravo and Mertikopoulos [25];

Learning vs. evolution under uncertainty

Now, since (SDA) boils down to the stochastic replicator dynamics (SRD) in the entropic case, Theorem 3.13 should be contrasted to the corresponding results of Cabrales [33] and Imhof [71] for (ASRD). Importantly, even though (SRD) and (ASRD) coincide in the noise-free case $\sigma = 0$, the Itô correction to the drift is different in the two cases: in the former, it stems from the positive reinforcement of strategies that perform well under (SXL); in the latter, it stems from the assumption that the per capita growth of the a_i-th strategy is driven by the perturbed payoff process $u_{ia_i} + \sigma\, dW_{ia_i}$. These mechanisms are equivalent in the noise-free case, but not in the presence of uncertainty: the reinforcement model outlined above can detect differences between the payoff of two strategies that the biological model cannot (because there is no inherent scaling in the payoff aggregation variable Y_t) [96]. By this token, learning and evolution end up leading to fairly different outcomes under uncertainty.

Stochastic stability

A similar phenomenon arises when considering the long-term stability and equilibrium convergence properties of (SDA). To set the stage, we first define the notions of Lyapunov and asymptotic stability in a stochastic differential equation (SDE) context. Following Khasminskii [76], we have:

Definition 3.4. Fix $x^* \in \mathcal{X}$ and let $X_t = Q(Y_t)$ denote a solution orbit of (SDA). We will then say that:

1. x^* is *stochastically* (*Lyapunov*) *stable* under (SDA) if, for all $\varepsilon > 0$ and every neighborhood U of x^* in $\mathcal{X}$, there exists a neighborhood $U' \subseteq U$ of x^* such that

$$\mathbb{P}(X_t \in U \text{ for all } t \geq 0 \,|\, X_0 \in U') \geq 1 - \varepsilon. \tag{3.35}$$

2. x^* is *stochastically asymptotically stable* under (SDA) if it is stochastically stable and attracting; that is, for all $\varepsilon > 0$ and every neighborhood U of x^* in $\mathcal{X}$, there exists a neighborhood $U' \subseteq U$ of x^* such that

$$\mathbb{P}(X_t \in U \text{ for all } t \geq 0 \text{ and } \lim_{t\to\infty} X_t = x^* \,|\, X_0 \in U') \geq 1 - \varepsilon. \tag{3.36}$$

In the evolutionary setting of (ASRD), Imhof [71] and Hofbauer and Imhof [64] showed that strict Nash equilibria are stochastically asymptotically stable provided that the variability of the shocks is small enough. Remarkably, this "small noise" assumption is not needed under (SDA):

Equilibrium convergence and stability properties

Theorem 3.14 (Bravo and Mertikopoulos, 2017)**.** *Fix $x^* \in \mathcal{X}$ and let $X_t = Q(Y_t)$ be a solution orbit of* (SDA)*. Then:*

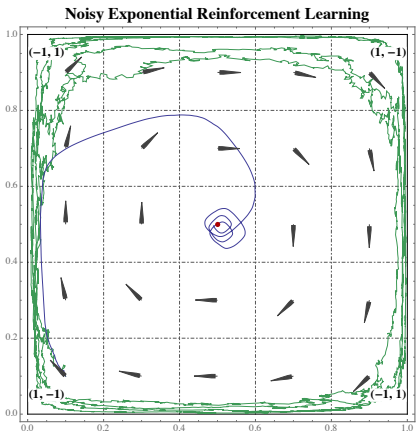
(1) *If $\mathbb{P}(\lim_{t\to\infty} X_t = x^*) > 0$, x^* is a Nash equilibrium.*

(2) *If x^* is stochastically* (*Lyapunov*) *stable, it is also Nash.*

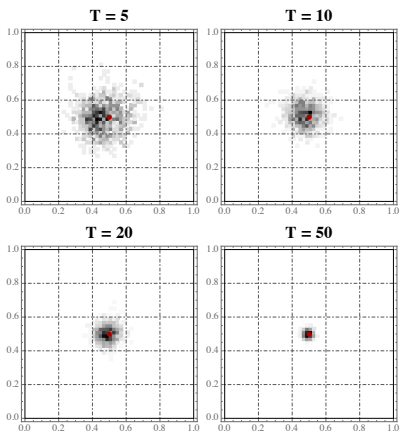
(3) *If x^* is a strict Nash equilibrium, it is also stochastically asymptotically stable.*

The importance of being strict

On the flip side of this theorem, if a Nash equilibrium $x^* \in \mathcal{X}$ is pure but not strict, then it cannot be attracting under (SDA) Heuristically, the reason for this is as follows:



(a) A sample trajectory and its time-average.



(b) Density of time averages at time T.

Figure 3.4: The long-run behavior of time-averaged trajectories of play under (SXL) in Matching Pennies. Fig. 3.4a shows the evolution of a sample trajectory and its time average; in Fig. 3.4b, we show a density plot of the distribution of 10^4 time-averaged trajectories for different values of the integration horizon T. As predicted by Proposition 3.16, time-averages converge to the game's Nash equilibrium.

if some player $i \in \mathcal{N}$ has two strategies $a_i, a_i' \in \mathcal{A}_i$ that give the same payoff against x_{-i}^*, the score difference $Y_{ia_i} - Y_{ia_i'}$ between the two will be dominated by the noise (since the drift vanishes). In this case, it is reasonable to expect that the dynamics (SDA) are attracted (locally and with high probability) to the face of $\mathcal{X}$ that is spanned by x^* and all best responses to x^*. However, a rigorous statement along these lines is fairly cumbersome to write down, so we do not provide one.

On the behavior of time-averages

As an alternative to the study of interior Nash equilibria, we analyze below the asymptotic behavior of the time-averaged process $\bar{X}_t$ of X_t in 2-player zero-sum games. Our analysis is motivated by the original deterministic results of Hofbauer and Sigmund [67] and Hofbauer et al. [69] who showed that $\bar{X}_t$ converges to Nash equilibrium under the (deterministic) replicator dynamics (RD), provided that $\liminf_{t\to\infty} X_{ia_i,t} > 0$ for all $a_i \in \mathcal{A}_i$, $i = 1, 2$. Our main contribution here is that the averaging principle of Hofbauer and Sigmund [67] extends to the stochastic dynamics (SDA), irrespective of the magnitude of the noise:

Averaging under permanency

Proposition 3.15. *Let Γ be a finite 2-player zero-sum game and let X_t be a solution orbit of the stochastic dynamics* (SDA)*. If the players' score differences $Y_{ia_i}(t) - Y_{ia_i'}(t)$ grow sublinearly for all $a_i, a_i' \in \mathcal{A}_i$, $a = 1, 2$, then the time-averaged process $\bar{X}_t$ converges almost surely to the Nash set of Γ.*

In the case of (XL)/(RD), the sublinear growth requirement for $Y_{ia_i} - Y_{ia_i'}$ boils down to the permanency condition $\liminf_{t\to\infty} X_{ia_i}(t) > 0$, so we recover the original result of Hofbauer and Sigmund [67]. However, the applicability of Proposition 3.15 is limited by the growth requirement for $Y_{ia_i} - Y_{ia_i'}$. The following proposition shows that this condition always holds in 2-player zero-sum games with an interior equilibrium:

Proposition 3.16 (Bravo and Mertikopoulos, 2017)**.** *Let Γ be a 2-player zero-sum game with an interior Nash equilibrium, and assume that* (SDA) *is run with a vanishing learning rate η_t satisfying Eq.* (3.25)*. Then, with probability* 1*, the time-average $\bar{X}_t$ of X_t converges to the set of Nash equilibria of Γ.*

The best response dynamics

In a certain sense, Proposition 3.16 is reminiscent of Theorem 3.8 on the no-regret properties of (SDA). This link between time-averaged orbits and the induced regret was the starting observation of Hofbauer et al. [69] who used it to derive a general averaging

principle linking the time-averaged behavior of (CDA) to the best response dynamics of Gilboa and Matsui [55], viz.

$$\dot{X}_i \in \mathrm{br}_i(X) - X_i, \tag{BRD}$$

where

$$\mathrm{br}_i(x) \equiv \operatorname*{arg\,max}_{x_i' \in \mathcal{X}_i} u_i(x_i'; x_{-i}) \tag{3.37}$$

denotes the *best response* (BR) correspondence of player i.

More precisely, Hofbauer et al. [69] showed that the ω-limit set Ω of the time-averages of (RD) is an *internally chain transitive* (ICT) set of (BRD), i.e., any two points in Ω can be joined by a piecewise continuous "chain" of arbitrarily long orbit segments of (BRD) lying in Ω up to arbitrarily small jump discontinuities.[10] As we show below, the *stochastic* dynamics (SDA) share the same connection to the *deterministic* best response dynamics (BRD), irrespective of the noise level:

The link between (SDA) *and* (BRD)

Theorem 3.17. *Let X_t be a solution orbit of* (SDA) *for a finite 2-player game Γ. Then, the ω-limit set of $\bar{X}_t$ is internally chain transitive under* (BRD).

Owing to Theorem 3.17, several conclusions of Hofbauer et al. [69] for 2-player games can be extended to the *stochastic* no-regret setting of (SDA) simply by exploiting the properties of the *deterministic* dynamics (BRD). Specifically, we obtain the following immediate corollaries of Theorem 3.17:

1. If $\bar{X}_t$ converges under (SDA), its limit is a Nash equilibrium.
2. Any global attractor of (BRD) also attracts the time-averaged orbits of (SDA), independently of the noise level. In particular, since the Nash set of a 2-player zero-sum game is globally attracting under (BRD), this observation extends Proposition 3.16 to the constant η case.
3. The only ICT sets of (BRD) in potential games consist of (isolated) components of Nash equilibria; hence, $\bar{X}_t$ converges to a component of $\mathrm{NE}(\Gamma)$.

RESULTS FOR CONCAVE GAMES. We now turn to games with continuous action spaces – and, in particular, monotone games. A first observation in this context is that the trajectory of play induced by (SDA) may fail to converge with probability 1, even in very simple games. For a concrete example, consider a single player with action space $\mathcal{X} = [-1, 1]$ and payoff function $u(x) = 1 - x^2/2$. Then, $V(x) = \nabla u(x) = -x$ for all $x \in [-1, 1]$, so (SDA) takes the form

Non-convergence with probability 1

$$\begin{aligned} dY &= -X_t\, dt + \sigma\, dW_t, \\ X_t &= [Y_t]_{-1}^{1}, \end{aligned} \tag{3.38}$$

where, for simplicity, we took σ to be constant, $\eta = 1$, and we used the shorthand $[x]_a^b$ for x if $x \in [a, b]$, a if $x \le a$, and b if $x \ge b$. In this case, the game's unique Nash equilibrium obviously corresponds to $x = 0 = [0]_0^1$. However, the dynamics (3.38) describe a truncated Orstein–Uhlenbeck (OU) process [74], leading to the explicit solution

$$Y_t = C_0 e^{-t} + \sigma \int_{t_0}^{t} e^{-(t-s)}\, dW_s \quad \text{for some } C_0 \in \mathbb{R}, \tag{3.39}$$

valid whenever $Y_s \in [-1, 1]$ for $s \in [t_0, t]$.

10 In particular, ICT sets are invariant, connected and have no proper attractors; for a full development, see Benaïm [14], Benaïm et al. [16], and refferences therein.

Thanks to this expression, several conclusions can be drawn regarding (3.38). First, even though the drift of the dynamics (3.38) vanishes at 0, the martingale part dW_t does not, so the process X_t *cannot converge to* 0 with positive probability, even when σ is arbitrarily small. Instead, X_t converges in distribution to a truncated Gaussian random variable X_∞ with mean 0 and variance proportional to σ^2 [74, Chap. 5.6]. Thus, in the long run, X_t will fluctuate around 0 with a spread that grows with the noise volatility coefficient σ. Furthermore, by ergodicity, the same is true for the time-averaged process $\bar{X}_t = (1/t)\int_0^t X_s\, ds$, i.e., the long-run average of X_t also fails to converge to equilibrium with positive probability.

To circumvent this negative result, we begin with the case where the players' gradient feedback becomes more accurate as measurements accrue over time – for instance, thanks to a variance reduction scheme or as in applications to wireless communications where the accumulation of pilot signals allows users to better sense their channel over time [100]. Our main result in this context is as follows:

Convergence under vanishing noise

Theorem 3.18 (Mertikopoulos and Staudigl, 2017)**.** *Let $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ be a strictly monotone game, let $X_t = Q(Y_t)$ be an orbit of* (SDA)*, and suppose that Assumptions 3.2 and 3.3 hold. If $\max_{x\in\mathcal{X}} \|\sigma(x,t)\| = o(1/\sqrt{\log t})$ as $t \to \infty$, X_t converges to the game's* (*necessarily unique*) *Nash equilibrium with probability* 1*.*

Heuristically, Theorem 3.18 provides an upper bound for the rate at which the noise must vanish so that convergence may arise in the long run; in particular, any power law decay rate is sufficient in that regard. Beyond this "vanishing noise", convergence under persistent noise does not seem possible. [11]

On the other hand, if we focus on the time-averages of (SDA), we have the following result:

Cesàro convergence in monotone games

Theorem 3.19 (Mertikopoulos and Staudigl, 2017)**.** *Let $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ be a monotone game satisfying Assumption 3.2, and let $X_t = Q(Y_t)$ be an orbit of* (SDA)*. Then, with probability* 1*, the time-average $\bar{X}_t$ of X_t enjoys the equilibrium convergence guarantee*

$$\mathrm{NI}(\bar{X}_t) \leq \frac{H}{\eta_t t} + \frac{\sigma_*^2}{2Kt}\int_0^t \eta_s\, ds + \mathcal{O}(t^{-1/2}\log\log t), \tag{3.40}$$

where

$$\mathrm{NI}(x) = \sum_{i\in\mathcal{N}}\left[\max_{x_i'\in\mathcal{X}_i} u_i(x_i'; x_{-i}) - u_i(x)\right] \tag{3.41}$$

denotes the game's Nikaido–Isoda (NI) *function. In particular, if $\eta_t \to 0$ as $t \to \infty$, the time-averaged trajectories of* (SDA) *converge to* $\mathrm{NE}(\mathcal{G})$ *with probability* 1*.*

Remark. The Nikaido–Isoda function was introduced in [112] and has the key property that

$$\mathrm{NI}(x) \geq 0 \quad \text{for all } x \in \mathcal{X} \tag{3.42}$$

with equality if and only if $x \in \mathrm{NE}(\mathcal{G})$. In this way, $\mathrm{NI}(x)$ is a natural figure of merit for testing the convergence of a given sequence to a Nash equilibrium (or, more precisely, the set thereof).

The bound (3.41) is formally similar to the value convergence guarantee (3.29) for (static) convex minimization problems, so the same remarks apply. In particular, convergence requires a vanishing η_t but, at the same time, (3.25) requires that η_t does not decay too fast (so that $\lim_{t\to\infty}\eta_t t = 0$). As in the case of Theorem 3.8, the optimal Cesàro convergence rate for the Nikaido–Isoda function is with a schedule of the form $\eta_t \propto 1/\sqrt{t}$.

11 Except perhaps if a Nash equilibrium x^* is located at a corner of $\mathcal{X}$ – specifically, if $V(x^*)$ belongs to the topological interior of the polar cone $\mathrm{PC}(x^*)$ to $\mathcal{X}$ at x^*. We examine this "sharpness" condition in Chapter 4.

4

LEARNING IN GAMES: ALGORITHMIC ANALYSIS

The continuous-time analysis of the previous chapter provides a basic chassis for understanding the properties of no-regret learning in games. At the same time however, it also highlights some inherent differences between continuous and discrete time – for instance, the gap between the $\mathcal{O}(\sqrt{T})$ minimax regret bound in discrete time vs. the $\mathcal{O}(1)$ regret achieved by online mirror descent in continuous time. In this chapter, we use these insights as a rough roadmap of what to expect in a bona fide algorithmic setting.

For concreteness, we will focus throughout on moderate-to-low feedback environments where a full function oracle is not available (or is otherwise impractical to access). In view of this, our main point of interest will be no-regret procedures induced by the lazy mirror descent / dual averaging algorithm:

$$\begin{aligned} Y_{t+1} &= Y_t + \gamma_t V_t \\ X_{t+1} &= Q(Y_{t+1}) \end{aligned} \tag{DA}$$

where the sequence of gradient signals V_t, $t = 1, 2, \dots$, is generated by a stochastic first-order oracle as follows:

1. In the *unilateral setting*
$$V_t = -[\nabla \ell_t(X_t) + Z_t] \tag{4.1}$$
where $\ell_t \colon \mathcal{X} \to \mathbb{R}$ is the sequence of loss functions encountered by the agent.

2. In the *multi-agent setting*
$$V_t = V(X_t) + Z_t \tag{4.2}$$
where $V(x) = (V_i(x))_{i \in \mathcal{N}}$ denotes the players' individual payoff gradient profile in a continuous game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$.

In both cases, we will assume that V_t satisfies the blanket SFO requirements (2.24), with the obvious substitution $\nabla_t \leftarrow -V_t$ for (2.24c). Towards the end of this chapter, we will also discuss the case of zeroth-order feedback (i.e., when each agent only has access to their individual payoff/loss) and other learning impediments.

4.1 NO-REGRET VS. CONVERGENCE: A DISCRETE-TIME REDUX

 # This section summarizes results from [80, 101, 102]

4.1.1 *Regret minimization*

A continuous-time approach to regret minimization

We first discuss a way of bounding the regret of (DA) based on the continuous-time guarantee (3.1) established in the previous chapter. For simplicity, we will consider online linear optimization problems with perfect oracle feedback, but the general case is not different.

To carry out this analysis, let $v_t \in \mathcal{V}^*$, $t = 1, 2, \dots$, be a sequence of payoff vectors as in Section 2.1, and let

$$V_t^c = v_{\lceil t \rceil} \quad \text{for all } t \geq 0 \tag{4.3}$$

denote the piecewise constant interpolation of v_t to continuous time. Then, to compare the continuous- and discrete-time regimes, we will write X_t^c for the continuous-time policy induced by (CDA) against V_t^c, and X_t^d for the sequence of actions generated by (DA) with (perfect) oracle feedback $V_t^d = v_t$. Dually to the above, we also let

$$Y_t^c = \int_0^t V_s^c \, ds \tag{4.4a}$$

and

$$Y_t^d = \sum_{s=1}^{t} V_t^d \tag{4.4b}$$

denote the corresponding gradient aggregation variables of (CDA) and (DA) respectively.

The discretization error

Now, assuming $\|v_t\|_* \leq L$ for all $t = 1, 2, \dots$ (cf. Assumption 2.2), the induced difference in payoffs can be expressed for all $\tau \in (0, 1)$ and all $t = 1, 2, \dots$ as

$$|\langle V_{t-\tau}^c, X_{t-\tau}^c \rangle - \langle v_t^d, X_t^d \rangle| = |\langle v_t, X_{t-\tau}^c - X_t^d \rangle| \leq L \|X_{t-\tau}^c - X_t^d\| \tag{4.5}$$

Moreoever, since h is K-strongly convex (so the mirror map Q is $(1/K)$-Lipschitz continuous), Eqs. (4.4a) and (4.4b) provide the discretization error bound

$$\|X_{t-\tau}^c - X_t^d\| \leq \frac{1}{K} \|Y_{t-\tau}^c - Y_t^d\|_* \leq \frac{1}{K} \int_{t-\tau}^{t} \|V_s\|_* \, ds \leq \frac{L\tau}{K}. \tag{4.6}$$

Hence, combining all of the above, the difference in the incurred regret can be expressed as

$$\begin{aligned} \left|\mathrm{Reg}^c(T) - \mathrm{Reg}^d(T)\right| &= \left| \int_0^T \langle V_t^c, X_t^c \rangle \, dt - \sum_{t=1}^{T} \langle v_t, X_t^d \rangle \right| \\ &\leq \sum_{t=1}^{T} \int_{t-1}^{t} |\langle V_{t-\tau}^c, X_{t-\tau}^c - X_t^d \rangle| \, d\tau \\ &\leq \sum_{t=1}^{T} L \int_{t-1}^{t} \frac{L\tau}{K} \, d\tau = \frac{L^2 T}{2K}. \end{aligned} \tag{4.7}$$

Continuous to discrete

Finally, recalling the bound $\mathrm{Reg}^c(T) \leq H$ of Theorem 3.2 and rescaling $V_t \leftarrow \gamma V_t$ to take into account the step-size of (DA), we obtain the discrete-time guarantee

$$\mathrm{Reg}(T) \equiv \mathrm{Reg}^d(T) \leq \frac{H}{\gamma} + \frac{\gamma L^2 T}{2K}. \tag{4.8}$$

This approach dates back to Sorin [139] who used it to derive the no-regret properties of the exponential weights algorithm in continuous and discrete time. Kwon and Mertikopoulos [80] subsequently extended the discretization analysis of Sorin [139] to online convex optimization problems and obtained the general bound (4.8) for (DA). Remarkably, this bound coincides with the bound (2.51) of Theorem 2.4 and can be easily extended to cover the more general oracle assumptions therein.

A key observation from the above is that the second term in (4.8) can be interpreted as the aggregation of T discretization errors, each of size $\mathcal{O}(\gamma L^2/K)$. Accordingly, we observe the following trade-off between continuous and discrete time: a larger step-size leads to "faster" regret minimization in continuous time, as measured by the H/γ term above; however, it also leads to a commensurately larger discretization error, as measured

by the term $\gamma L^2 T/(2/K)$. Obtaining optimal rates in discrete time requires balancing these two terms, but the impact of discretization cannot be eliminated.

4.1.2 *Limit cycles and persistence of off-equilibrium behavior*

The discretization analysis of the previous section suggests that the discrete-time equilibrium convergence properties of (DA) cannot be better than their continuous-time counterparts. Thus, in view of the recurrence properties of the continuous-time dynamics (CDA), it stands to reason that the discrete-time algorithm (DA) would also fail to converge to Nash equilibrium in zero-sum games. We examine this question below in the context of continuous saddle-point problems of the form

$$\min_{x_1 \in \mathcal{X}_1} \max_{x_2 \in \mathcal{X}_2} \Phi(x_1, x_2), \tag{SP}$$

where each player's action space $\mathcal{X}_i$, $i = 1, 2$, is a closed convex subset of a finite-dimensional normed space $\mathcal{V}_i \equiv \mathbb{R}^{n_i}$, and $\Phi: \mathcal{X} \equiv \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ denotes the problem's value function. Clearly, the finite game framework of Section 3.2 is recovered when $\mathcal{X}_i = \Delta(\mathcal{A}_i)$ for a finite set of pure strategies $\mathcal{A}_i$, $i = 1, 2$, and Φ is bilinear over $\mathcal{X}_1 \times \mathcal{X}_2$.

Most of the literature on saddle-point problems has focused on the *monotone* case, i.e., when Φ is convex-concave. As we discussed in Section 2.3.2, Nash equilibria can then be characterized as solutions to the variational inequality

$$\langle V(x^*), x - x^* \rangle \leq 0 \quad \text{for all } x \in \mathcal{X}, \tag{VI}$$

where

$$V(x) = (-\nabla_{x_1} \Phi(x), \nabla_{x_2} \Phi(x)) \tag{4.9}$$

denotes the individual payoff gradient field of Φ. When Φ is convex-concave, this is in turn equivalent to solving the *Minty* variational inequality

Minty variational characterization

$$\langle V(x), x - x^* \rangle \leq 0 \quad \text{for all } x \in \mathcal{X}. \tag{MVI}$$

Importantly, this equivalence extends beyond the realm of (pseudo-)convex-concave problems. For a concrete non-monotone example, consider the problem

$$\min_{x_1 \in [-1,1]} \max_{x_2 \in [-1,1]} (x_1^4 x_2^2 + x_1^2 + 1)(x_1^2 x_2^4 - x_2^2 + 1). \tag{4.10}$$

A straightforward calculation shows that only saddle-point of Φ is $x^* = (0, 0)$: it is easy to check that x^* is also the unique solution of the corresponding problem (MVI), despite the fact that Φ is not even (quasi-)monotone.[1] This shows that the equivalence between (SP) and (MVI) encompasses cases that are incompatible with convexity/monotonicity assumptions, even in the lowest possible dimension; for an in-depth discussion of the links between (SP) and (MVI), we refer the reader to [50].

Motivated by this equivalence, we introduce below the notion of *coherence:*

Definition 4.1. We say that (SP) is *coherent* if every saddle-point of Φ is a solution of (MVI) and vice versa. If (MVI) holds as a strict inequality when x is not a saddle-point of Φ, (SP) will be called *strictly coherent*; by contrast, if (MVI) holds as an equality for all $x \in \mathcal{X}$, we will say that (SP) is *null-coherent*.

Coherence

Remark 4.1. To the best of the author's knowledge, the study of gradient conditions of this type can be traced back at tleast to [23]; the term "coherence" is borrowed from [102, 153]. We should also note that it is possible to relax the equivalence between (SP)

1 To see this, simply note that $\Phi(x_1, x_2)$ is *multi-modal* in x_2 for certain values of x_1.

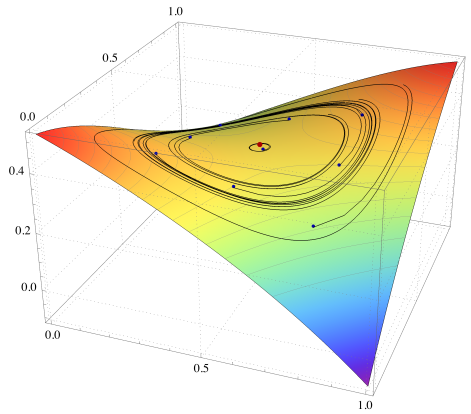


Figure 4.1: Non-convergence of (DA) in the non-monotone saddle-point problem $\Phi(x_1, x_2) = (x_1 - 1/2)(x_2 - 1/2) + \frac{1}{3}\exp(-(x_1 - 1/4)^2 - (x_2 - 3/4)^2)$.

and (MVI) by positing that only *some* of the solutions of (SP) can be harvested from (MVI). For simplicity, we do not pursue this relaxation here.

Remark 4.2. As an example, if Φ is strongly convex-concave, (SP) is strictly coherent. By contrast, 2-person finite zero-sum games with an interior equilibrium are null-coherent [102]. In view of this connection, we will focus here on null-coherent games; strictly coherent games will be studied in detail in the next section.

Now, going back to the analysis of Section 3.2, the key to establishing the recurrence properties of the continuous-time dynamics (CDA) was the Fenchel coupling

$$F(x, y) = h(x) + h^*(y) - \langle y, x\rangle \quad \text{for all } x \in \mathcal{X},\ y \in \mathcal{Y}. \tag{4.11}$$

As we discussed in Section 3.2, if x^* is an interior equilibrium of a finite zero-sum game, $F(x^*, X_t)$ is a constant of motion under the dynamics (CDA). In the case of the discrete-time system (DA), this fragile property is replaced by the following asymptotic variant:

Non-convergence in zero-sum games

Theorem 4.1 (Mertikopoulos et al., 2019)**.** *Suppose that* (SP) *is null-coherent and* (DA) *is run with unbiased first-order oracle feedback of the form* (2.31)*. Then, for every Nash equilibrium x^* of* (SP)*, we have:*

a) The sequence $\mathbb{E}[F(x^*, Y_t)]$ is non-decreasing.

b) If, in addition, $\sum_t \gamma_t^2 < \infty$, the sequence $F(x^*, Y_t)$ converges (a.s.) to a random variable F_∞ with $\mathbb{E}[F_\infty] < \infty$.

Corollary 4.2. *Suppose that Φ is bilinear and admits an interior equilibrium $x^* \in \operatorname{ri}(\mathcal{X})$. If $X_1 \neq x^*$ and* (DA) *is run with a perfect gradient oracle (i.e., $B = 0$ and $\sigma = 0$), we have $\lim_{t\to\infty} D(x^*, X_t) > 0$.*

In words, the above shows that (DA) does not converge in finite zero-sum games with a unique interior equilibrium: instead, the induced sequence of play cycles at positive Bregman divergence from the game's Nash equilibrium. Heuristically, the reason for this behavior is that, for small $\gamma \to 0$, the incremental step $\gamma V(Q(y))$ of (DA) is essentially tangent to the level set of $F(x^*, \cdot)$ that passes through y. For finite $\gamma > 0$, things are even worse because this increment points noticeably away from y, i.e., towards higher level sets of F. As a result, the "best-case scenario" for (DA) is to orbit x^* (when $\gamma \to 0$); in practice, for finite γ, the algorithm takes small outward steps throughout its runtime, eventually converging to some limit cycle farther away from x^*.

This failure of (DA) is due to the fact that, witout a mitigating mechanism in place, a "blind" first-order step could overshoot and lead to an outwards spiral, even with a vanishing step-size. We will revisit this issue towards the end of this chapter; for a numerical illustration, see Fig. 4.1.

4.2 CONVERGENCE TO EQUILIBRIUM AND RATIONALIZABILITY

 # This section summarizes results from [40, 97, 102]

We now proceed to establish a series of equilibrium convergence and rationalizability results for (DA). To connect our discussion with that of the previous section, we first present a series of results for concave games and then examine finite games towards the end of this section.

4.2.1 *Positive results in concave games*

VARIATIONAL STABILITY. We begin by revisiting the notion of coherence and the variational characterization (MVI) of Nash equilibria in monotone games. Specifically, (MVI) states that the players' individual payoff gradients point (weakly) "towards" the Nash set of a monotone game $\mathcal{G}$ in the sense that $V(x)$ forms an acute angle with $x^* - x$ for all $x^* \in \mathcal{X}^* \equiv \mathrm{NE}(\mathcal{G})$. This observation motivates the following definition:

Variational stability

Definition 4.2. We say that $x^* \in \mathcal{X}$ is *variationally stable* (or simply *stable*) if there exists a neighborhood U of x^* in $\mathcal{X}$ such that

$$\langle V(x), x - x^* \rangle \leq 0 \quad \text{for all } x \in U,$$

with equality if and only if $x = x^*$. In particular, if U can be taken to be all of $\mathcal{X}$, we say that x^* is *globally variationally stable* (or *globally stable* for short).

Remark 4.3. The terminology "variational stability" alludes to the seminal notion of *evolutionary stability* (ES) introduced by Maynard Smith and Price [88] for population games (i.e., games with a continuum of players and a common, finite set of actions $\mathcal{A}$). Specifically, if $V(x) = (u_1(x), \dots, u_N(x))$ denotes the payoff field of such a game (with $x \in \Delta(\mathcal{A})$ denoting the state of the population and $u_a(x)$ denoting the reproductive fitness of the a-th genotype at state x), Definition 4.3 boils down to the variational characterization of evolutionarily stable states by Taylor [142] and Hofbauer et al. [68]. As we show in the sequel, variational stability plays the same role for learning in games with continuous action spaces as evolutionary stability plays for evolution in games with a continuum of players.

Variational stability vs. monotonicity

By definition, if a continuous game $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ is strictly monotone, its (unique) Nash equilibrium is globally stable; the converese however does not hold, even partially. As an example, consider the single-player game with payoffs given by the function

$$u(x) = 1 - \sum_{\ell=1}^{n} \sqrt{1 + x_\ell}, \quad x \in [0,1]^n. \tag{4.12}$$

In this simple example, the origin is the unique maximizer of u (and hence the game's unique Nash equilibrium). Moreover, we trivially have $\langle V(x), x \rangle = -2 \sum_{\ell=1}^{n} x_\ell / \sqrt{1 + x_\ell} \leq 0$ with equality if and only if $x = 0$, so the origin satisfies the global version of (VS); however, u is not even pseudo-concave if $n \geq 2$, so the game cannot be monotone.

In words, strict monotonicity is a sufficient condition for the existence of a (globally) stable state, but not a necessary one. That being said, even in this (non-monotone) example, variational stability characterizes the game's unique Nash equilibrium. We make this link precise below:

Proposition 4.3 (Mertikopoulos and Zhou, 2019)**.** *Suppose that $x^* \in \mathcal{X}$ is variationally stable. Then:*

a) *If $\mathcal{G}$ is concave, x^* is an isolated Nash equilibrium of $\mathcal{G}$.*

b) *If x^* is globally stable, it is the game's unique Nash equilibrium.*

Setwise notions of variational stability

Proposition 4.3 indicates that variationally stable states are isolated. However, this also means that Nash equilibria of games with a concave – but not *strictly* concave – potential may fail to be stable.[2] To account for such cases, we will also consider the following setwise stability notion:

Definition 4.3. Let $\mathcal{C} \subseteq \mathcal{X}$ be closed and nonempty. We say that $\mathcal{C}$ is *variationally stable* (or simply *stable*) if there exists a neighborhood U of $\mathcal{C}$ in $\mathcal{X}$ such that

$$\langle V(x), x - x^* \rangle \leq 0 \quad \text{for all } x^* \in \mathcal{C} \text{ and all } x \in U, \tag{VS}$$

with equality for a given $x^* \in \mathcal{C}$ if and only if $x \in \mathcal{C}$. In particular, if U can be taken to be all of $\mathcal{X}$, we say that $\mathcal{C}$ is *globally variationally stable* (or *globally stable* for short).

Obviously, Definition 4.3 subsumes Definition 4.2: if $x^* \in \mathcal{X}$ is stable in the pointwise sense of Definition 4.2, it is also stable when viewed as a singleton set. When this is the case, it is also easy to see that x^* cannot belong to some larger variationally stable set,[3] so the notion of variational stability tacitly implies a certain degree of maximality. This is made clearer in the following:

Proposition 4.4 (Mertikopoulos and Zhou, 2019)**.** *Suppose that $\mathcal{C} \subseteq \mathcal{X}$ is variationally stable. Then:*

a) *$\mathcal{C}$ is convex.*

b) *If $\mathcal{C}$ is globally stable, it coincides with the game's set of Nash equilibria.*

c) *If $\mathcal{G}$ is concave, $\mathcal{C}$ is an isolated component of Nash equilibria.*

d) *If $\mathcal{G}$ is strictly coherent, $\mathcal{C}$ is globally stable and it coincides with $\mathcal{X}^* \equiv \mathrm{NE}(\mathcal{G})$.*

A Hessian test for variational stability

A second-order test to verify whether (VS) holds can be stated via the game's *Hessian matrix*, defined here as the block matrix $H(x) = (H_{ij}(x))_{i,j\in\mathcal{N}}$ with

$$\begin{aligned} H_{ij}(x) &= \tfrac{1}{2}\nabla_{x_j}\nabla_{x_i}u_i(x) + \tfrac{1}{2}(\nabla_{x_i}\nabla_{x_j}u_j(x))^\top \\ &= \tfrac{1}{2}\nabla_{x_j}V_i(x) + \tfrac{1}{2}(\nabla_{x_i}V_j(x))^\top \end{aligned} \tag{4.13}$$

We then have the following test for variational stability:

Proposition 4.5 (Mertikopoulos and Zhou, 2019)**.** *If x^* is a Nash equilibrium of $\mathcal{G}$ and $H(x^*) \prec 0$ on $\mathrm{TC}(x^*)$, then x^* is stable – and hence an isolated Nash equilibrium. In particular, if $H(x) \prec 0$ on $\mathrm{TC}(x)$ for all $x \in \mathcal{X}$, x^* is globally stable – so it is the unique equilibrium of $\mathcal{G}$.*

CONVERGENCE ANALYSIS. In the rest of this section, we use the notion of variational stability to derive some general convergence results for the sequence of play induced by (DA). Specifically, we first show that if (DA) converges to some action profile, this limit is a Nash equilibrium; subsequently, we show that globally (resp. locally) stable states are globally (resp. locally) attracting under (DA).

We begin by showing that if the sequence of play induced by (DA) converges to some $x^* \in \mathcal{X}$ with positive probability, this limit is a Nash equilibrium:

2 Recall here that $\mathcal{G}$ is a potential game if it admits a potential function $F\colon \mathcal{X} \to \mathbb{R}$ such that $V = \nabla F$ [103, 128, 129]. We discuss potential games in more detail in the next section.

3 In that case (VS) would give $\langle V(x'), x' - x^* \rangle = 0$ for some $x' \neq x^*$, a contradiction.

Limit points of (DA)

Theorem 4.6 (Mertikopoulos and Zhou, 2019)**.** *Suppose that* (DA) *is run with unbiased first-order feedback of the form* (2.31) *and a step-size sequence γ_t such that*

$$\sum_{t=1}^{\infty}\left(\frac{\gamma_t}{\theta_t}\right)^2 < \sum_{t=1}^{\infty}\gamma_t = \infty, \tag{4.14}$$

where $\theta_t = \sum_{s=1}^{t}\gamma_s$. If the underlying game is concave and X_t converges to $x^* \in \mathcal{X}$ with positive probability, then x^* is a Nash equilibrium of $\mathcal{G}$.

Remark 4.4. Note here that the requirement (4.14) holds for every step-size policy of the form $\gamma_t \propto 1/t^b$, $b \leq 1$ (i.e. even for *increasing* γ_t).

We continue with a series of direct convergence results for (DA). Our analysis will be carried out under the Fenchel reciprocity condition (Assumption 3.3) which, for convenience, we recall below:

$$F(x, y_k) \to 0 \quad \text{whenever} \quad Q(y_k) \to x \tag{3.23}$$

for every sequence $y_k \in \mathcal{Y}$. Under this regularity assumption, we have:

Global convergence

Theorem 4.7 (Mertikopoulos and Zhou, 2019)**.** *Suppose that* (DA) *is run with unbiased first-order feedback of the form* (2.31) *and a step-size sequence γ_t such that $\sum_{t=1}^{\infty}\gamma_t^2 < \infty$ and $\sum_{t=1}^{\infty}\gamma_t = \infty$. If the game's Nash set is globally stable and Assumption 3.3 holds, the sequence of actions X_t generated by* (DA) *converges to a Nash equilibrium of $\mathcal{G}$* (*a.s.*).

Corollary 4.8. *If $\mathcal{G}$ is strictly monotone or strictly coherent, X_t converges to the game's* (*necessarily unique*) *Nash equilibrium with probability* 1.

The first step in the proof of Theorem 4.7 is to show that the sequence of dual states Y_t comprises an *asymptotic pseudotrajectory* (APT) of the continuous-time dynamics (CDA).[4] APTs have the key property that, in the presence of a global attractor, they cannot stray too far from the flow of the "mean field" dynamics (CDA). However, given that the mirror map Q may fail to be invertible, the standard theory of APTs does not apply to the primal state sequence $X_t = Q(Y_t)$ (and convergence to equilibrium cannot be studied in $\mathcal{Y}$ if $\mathcal{X}^* \subsetneq \mathcal{X}^\circ \equiv \operatorname{dom}\partial h$). Instead, the proof of Theorem 4.7 requires deriving a uniform bound for the convergence of the continuous-time dynamics (CDA) to an ε-neighborhood of $\mathcal{X}^* \equiv \mathrm{NE}(\mathcal{G})$, and an inductive shadowing argument to show that no APT generated in this way may escape from this neighborhood.

In terms of the algorithm's step-size, note that all policies of the form $\gamma_t \propto 1/t^b$, $b \in (1/2, 1]$ are allowed under Theorem 4.7. The "critical" value $b = 1/2$ is tied to the finite second-moment hypothesis (2.24c). If the players' gradient observations have finite moments up to some order $q > 2$, a more refined stochastic approximation argument can be used to show that Theorem 4.7 still holds under the lighter requirement $\sum_{t=1}^{\infty}\gamma_t^{1+q/2} < \infty$. Thus, even in the presence of noise, it is possible to employ (DA) with any step-size sequence of the form $\gamma_t \propto 1/t^b$, $b \in (0, 1]$, provided that the noise process U_t has $\mathbb{E}[\|U_t\|_*^q \mid \mathcal{F}_t] < \infty$ for some $q > 2/b - 2$. In particular, if the noise affecting the players' observations has finite moments of all orders (for instance, if U_t is sub-exponential or sub-Gaussian), it is possible to use any $b \in (0, 1]$.

We now proceed to show that (DA) remains locally convergent to states that are only locally stable with probability arbitrarily close to 1:

Local convergence

Theorem 4.9 (Mertikopoulos and Zhou, 2019)**.** *Fix a confidence level $\alpha > 0$ and suppose that* (DA) *is run with unbiased first-order feedback of the form* (2.31) *and a sufficiently*

4 Intuitively, this means that Y_t asymptotically tracks the flowlines of (CDA) with arbitrary accuracy over windows of arbitrary length. For a precise definition, see Benaïm [14], Benaïm and Hirsch [15] and Benaïm et al. [16].

small step-size γ_t satisfying $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ and $\sum_{t=1}^{\infty} \gamma_t = \infty$. If $\mathcal{C} \subseteq \mathcal{X}$ is stable and Assumption 3.3 holds, then $\mathcal{C}$ is locally attracting with probability at least $1 - \alpha$; more precisely, there exists a neighborhood U of $\mathcal{C}$ in $\mathcal{X}$ such that

$$\mathbb{P}(\lim_{t\to\infty} X_t \in \mathcal{C} \mid X_1 \in U) \geq 1 - \alpha. \tag{4.15}$$

Corollary 4.10. *Let x^* be a Nash equilibrium with $H(x^*) < 0$. Then, with assumptions as above, x^* is locally attracting with probability arbitrarily close to* 1.

Sharp equilibria

Because of the random shocks induced by the noise in the players' gradient observations, it is difficult to obtain an almost sure (or high probability) estimate for the convergence rate of the last iterate X_t of (DA). Specifically, even with a rapidly decreasing step-size policy, a single realization of the error process Z_t may lead to an arbitrarily big jump of X_t at any time, thus destroying any almost sure bound on the convergence rate of X_t. This obstacle can be overcome under the following condition:

Definition 4.4. We say that $x^* \in \mathcal{X}$ is a *sharp equilibrium* of $\mathcal{G}$ if

$$\langle V(x^*), z\rangle \leq 0 \quad \text{for all } z \in \mathrm{TC}(x^*), \tag{4.16}$$

with equality if and only if $z = 0$.

Remark 4.5. The terminology "sharp" follows Polyak [116, Chapter 5.2], who introduced a similar notion for (unconstrained) convex programs. In particular, in the single-player case, it is easy to see that (4.16) implies that x^* is a *sharp maximum* of $u(x)$, i.e. $u(x^*) - u(x) \geq \lambda\|x - x^*\|$ for some $\lambda > 0$. More generally, in *finite* games, Definition 4.4 is equivalent to the notion of a *strict* Nash equilibrium (cf. Proposition 4.13 below). The reason that we employ the terminology "sharp" instead of "strict" is that strict version of the Nash equilibrium inequality (NE) is a weaker requirement than Definition 4.4 if the game is not linear.

A first consequence of Definition 4.4 is that $V(x^*)$ lies in the topological interior of the polar cone $\mathrm{PC}(x^*)$ to $\mathcal{X}$ at x^*. In turn, this implies that sharp equilibria can only occur at *corners* of $\mathcal{X}$ (i.e., points whose polar cone has nonempty topological interior; for a schematic illustration, see Fig. 2.5). By continuity, this further implies that sharp equilibria are locally stable so, by Proposition 4.4, sharp equilibria are also isolated. Our next result shows that if players employ (DA) with surjective mirror maps, then, with high probability, sharp equilibria are attained in a *finite* number of steps:

Fast convergence to sharp equilibria

Theorem 4.11 (Mertikopoulos and Zhou, 2019)**.** *Fix a confidence level $\alpha > 0$ and suppose that* (DA) *is run with unbiased first-order feedback of the form* (2.31) *and a sufficiently small step-size γ_t satisfying $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ and $\sum_{t=1}^{\infty} \gamma_t = \infty$. If $x^* \in \operatorname{dom} \partial h$ is sharp, there exists a neighborhood U of x^* in $\mathcal{X}$ such that*

$$\mathbb{P}(X_t = x^* \text{ for all sufficiently large } t \mid X_1 \in U) \geq 1 - \alpha, \tag{4.17}$$

In particular, if x^* is globally stable, X_t converges to x^* in a finite number of steps from every initial condition (*a.s.*).

Theorem 4.11 suggests that dual averaging with surjective mirror maps ($\operatorname{im} Q = \mathcal{X}$ or, equivalently, $\mathcal{X}^\circ \equiv \operatorname{dom} \partial h = \mathcal{X}$) leads to significantly faster convergence to sharp equilibria. This is consistent with the observations made in Chapter 3 for the convergence of the continuous-time, deterministic dynamics (CDA) to strict equilibria finite games. We should also state here that this result does not hold for the eager variant of (DA): in that case, any "bad" realization of the oracle noise process Z_t can take X_t out of equilibrium. For a detailed statement in the context of nonlinear programming, we refer the reader to Zhou et al. [156].

4.2.2 *Positive results in finite games*

Dual averaging in finite games

We now turn to the analysis of (DA) in finite games. Concretely, given a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$, we will focus on the following discrete-time implementation of (DA): At each stage $t = 1, 2, \dots$, every player $i \in \mathcal{N}$ selects a pure strategy $\hat{a}_{i,t} \in \mathcal{A}_i$ according to their individual mixed strategy $X_{i,t} \in \mathcal{X}_i \equiv \Delta(\mathcal{A}_i)$. Subsequently, each player observes – or otherwise estimates – the payoffs of their pure strategies $a_i \in \mathcal{A}_i$ against the chosen actions $\hat{a}_{-i,t}$ of all other players (possibly subject to some random estimation error). Specifically, we posit that each player receives as feedback the noisy payoff signal

$$V_{ia_i,t} = u_i(a_i; \hat{a}_{-i,t}) + U_{ia_i,t}, \tag{4.18}$$

where the noise process $U_t = (U_{ia_i,t})_{a_i \in \mathcal{A}_i, i \in \mathcal{N}}$ is assumed to satisfy (2.24). Then, based on this feedback, players update their mixed strategies based on (DA) and the process repeats.

Our first result in this setting concerns the elimination of dominated strategies:

Elimination of dominated strategies

Theorem 4.12 (Cohen et al., 2017)**.** *Suppose that* (DA) *is run with noisy payoff observations of the form* (4.18) *and a step-size sequence γ_t satisfying* (4.14)*. If $a_i \in \mathcal{A}_i$ is dominated, then $X_{ia_i,t} \to 0$ with probability* 1*.*

This result can be seen as a direct analogue of Theorem 3.4 for the continuous-time dynamics (CDA). However, we should note here that the corresponding result does not hold for the eager variant of (DA): even if the eager version of the algorithm is initialized at a face of $\mathcal{X}$ where no dominated strategies are present, a single realization of the noise process U_t that inverts the payoff relation of two strategies could lead to dominated strategies remaining present (at least, infinitely often). This highlights yet another difference between the eager and lazy variants of (DA) in the context of game-theoretic learning (at least, when the algorithm is run with a nonsteep regularizer for which $\operatorname{dom} \partial h = \mathcal{X}$).

Convergence to strict Nash equilibria

We now proceed to the question of convergence to strict Nash equilibria: to begin, recall that a Nash equilibrium x^* of a finite game is called *strict* when (NE) holds as a strict inequality for all $x_i \neq x_i^*$. This implies that strict Nash equilibria are pure strategy profiles $x^* = (a_1^*, \dots, a_N^*)$ such that

$$u_i(a_i^*; a_{-i}^*) > u_i(a_i; a_{-i}^*) \quad \text{for all } a_i \in \mathcal{A}_i \setminus \{a_i^*\},\ i \in \mathcal{N}. \tag{4.19}$$

In fact, strict Nash equilibria of a finite game $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ can be characterized further as follows:

Strict equilibria are sharp

Proposition 4.13. *The following are equivalent:*

a) *x^* is a strict Nash equilibrium of* Γ*.*

b) *$\langle V(x^*), z \rangle \leq -\lambda z$ for some $\lambda > 0$ and all $z \in \operatorname{TC}(x^*)$.*

c) *x^* is variationally stable.*

d) *x^* is sharp.*

Thanks to the above characterization of strict equilibria, the convergence analysis of the previous section yields:

Convergence to strict equilibria

Proposition 4.14 (Mertikopoulos and Zhou, 2019)**.** *Let x^* be a strict Nash equilibrium of a finite game $\mathcal{G}$ and fix a confidence level $\alpha > 0$. Suppose further that* (DA) *is run with noisy payoff observations of the form* (4.18) *and a sufficiently small step-size γ_t such that $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ and $\sum_{t=1}^{\infty} \gamma_t = \infty$. Then:*

1. *If $x^* \in \operatorname{dom} \partial h$, there exists a neighborhood U of x^* such that*

$$\mathbb{P}(X_t = x^* \text{ for all sufficiently large } t \mid X_1 \in U) \geq 1 - \alpha. \tag{4.20}$$

2. *If $x^* \notin \operatorname{dom} \partial h$ and the Fenchel reciprocity condition* (3.23) *holds, there exists a neighborhood U of x^* such that*

$$\mathbb{P}(\lim_{t\to\infty} X_t = x^* \mid X_1 \in U) \geq 1 - \alpha. \tag{4.21}$$

Cohen et al. [40] proved a special case of this result for (Hedge) and further showed that the algorithm's convergence rate is exponential in the "running horizon" $\theta_t = \sum_{s=1}^{t} \gamma_s$. This rate is closely linked to the logit choice model, and different mirror maps yield different convergence speeds; for a detailed discussion, we refer the reader to [97].

4.3 LEARNING WITH BANDIT FEEDBACK

 # This section summarizes results from [26, 39]

Bandit feedback

In this section, we drop the stochastic first-order oracle feedback requirement, and we focus on the *bandit feedback* case, i.e., when the only information at the players' disposal is the payoffs they receive at each stage. For obvious reasons, this extra degree of uncertainty complicates matters considerably because players must now estimate their payoff gradients from their observed rewards. What makes matters even worse is that an agent may introduce a non-negligible bias in the (concurrent) gradient estimation process of another through the co-dependence of the players' payoff functions. As a result, conventional multiple-point estimation techniques for derivative-free optimization cannot be applied (at least, not without significant communication overhead between players). To do away with such coordination requirements, we focus on learning processes which could be sensibly deployed in a multi-agent setting and we explore the resulting equilibrium convergence properties.

4.3.1 *Payoff-based learning in concave games*

To begin with the obvious, if players don't have access to a first-order oracle, they will need to construct one from the only information at their disposal: the actual payoffs they receive at each stage. When a function can be queried at multiple points (in practice, as few as two), there are efficient ways to estimate its gradient via directional sampling techniques as in [2]. In a game-theoretic setting however, multiple-point estimation techniques do not apply because, in general, a player's payoff function depends on the actions of *all* players. Thus, when a player attempts to get a second query of their payoff function, this function may have already changed because of the action taken by another player – i.e., instead of sampling $u_i(\cdot; x_{-i})$, the i-th player would be sampling $u_i(\cdot; x'_{-i})$ for some $x'_{-i} \neq x_{-i}$.

Simultaneous perturbation stochastic approximation

Following Spall [141] and Flaxman et al. [51], we posit instead that players rely on a *simultaneous perturbation stochastic approximation* (SPSA) approach that allows them to estimate their individual payoff gradients V_i based off a *single* function evaluation. In detail, the key steps of this one-shot estimation process for each player $i \in \mathcal{N}$ are:

1. Fix a *query radius* $\delta > 0$.[5]
2. Pick a *pivot point* $x_i \in \mathcal{X}_i$ where player i seeks to estimate their payoff gradient.
3. Draw a vector z_i from the unit sphere $\mathbb{S}_i \equiv \mathbb{S}^{n_i}$ of $\mathcal{V}_i \equiv \mathbb{R}^{n_i}$ and play $\hat{x}_i = x_i + \delta z_i$.[6]

5 For simplicity, we take δ equal for all players; the extension to player-specific δ is straightforward.

6 We tacitly assume here that the query directions $z_i \in \mathbb{S}^{n_i}$ are drawn independently across players.

4. Receive $\hat{u}_i = u_i(\hat{x}_i; \hat{x}_{-i})$ and define the SPSA oracle

$$\hat{v}_i = \frac{n_i}{\delta} \hat{u}_i \, z_i. \tag{SPSA}$$

By adapting a standard argument based on Stokes' theorem [26], it can be shown that $\hat{v}_i$ is an unbiased estimator of the individual gradient of the δ-smoothed payoff function

$$u_i^\delta(x) = \frac{1}{\operatorname{vol}(\delta\mathbb{B}_i) \prod_{j \neq i} \operatorname{vol}(\delta\mathbb{S}_j)} \int_{\delta\mathbb{B}_i} \int_{\prod_{j \neq i} \delta\mathbb{S}_j} u_i(x_i + w_i; x_{-i} + z_{-i}) \, dz_1 \cdots dw_i \cdots dz_N \tag{4.22}$$

with $\mathbb{B}_i \equiv \mathbb{B}^{n_i}$ denoting the unit ball of $\mathcal{V}_i$.[7] If V_i is Lipschitz continuous, then $\| \nabla_i u_i - \nabla_i u_i^\delta \|_\infty = \mathcal{O}(\delta)$, so this estimate becomes more and more accurate as $\delta \to 0^+$. On the other hand, the second moment of $\hat{v}_i$ grows as

$$\mathbb{E}[\|\hat{v}_i\|^2] = \frac{n_i^2}{\delta^2} \mathbb{E}[\|\hat{u}_i z_i\|^2] = \mathcal{O}(1/\delta^2), \tag{4.23}$$

implying in turn that the variability of $\hat{v}_i$ grows unbounded as $\delta \to 0^+$. This manifestation of the bias-variance dilemma plays a crucial role in designing no-regret policies with bandit feedback [51, 79], so δ must be chosen with care.

Feasibility issues

However, before dealing with this choice, it is important to highlight two feasibility issues that arise with the single-shot SPSA estimate (SPSA). The first has to do with the fact that the perturbation direction z_i is chosen from the unit sphere $\mathbb{S}_i$ so it may fail to be tangent to $\mathcal{X}_i$, even when x_i is interior. To iron out this wrinkle, it suffices to sample z_i from the intersection of $\mathbb{S}_i$ with the affine hull of $\mathcal{X}_i$ in $\mathcal{V}_i$; on that account (and without loss of generality), we will assume in what follows that each $\mathcal{X}_i$ is a *convex body* of $\mathcal{V}_i$, i.e., it has nonempty topological interior.

The second feasibility issue concerns the size of the perturbation step: even if z_i is a feasible direction of motion, the query point $\hat{x}_i = x_i + \delta z_i$ may be unfeasible if x_i is too close to the boundary of $\mathcal{X}_i$. For this reason, we will introduce a "safety net" in the spirit of Bubeck and Cesa-Bianchi [30], and we will constrain the set of possible pivot points x_i to lie within a suitably "deflated" zone of $\mathcal{X}$.

A safety net for sampling

To make this precise, let $\mathbb{B}_{R_i}(p_i)$ be an R_i-ball centered at $p_i \in \mathcal{X}_i$ so that $\mathbb{B}_{R_i}(p_i) \subseteq \mathcal{X}_i$. Then, instead of perturbing x_i by z_i, we consider the *feasibility adjustment*

$$w_i = z_i - R_i^{-1}(x_i - p_i), \tag{4.24}$$

and each player plays $\hat{x}_i = x_i + \delta w_i$ instead of $x_i + \delta z_i$. In other words, this adjustment moves each pivot to $x_i^\delta = x_i - R_i^{-1}\delta(x_i - p_i)$, i.e., $\mathcal{O}(\delta)$-closer to the interior base point p_i, and then perturbs x_i^δ by δz_i. Feasibility of the query point is then ensured by noting that

$$\hat{x}_i = x_i^\delta + \delta z_i = (1 - R_i^{-1}\delta) x_i + R_i^{-1}\delta(p_i + R_i z_i), \tag{4.25}$$

so $\hat{x}_i \in \mathcal{X}_i$ whenever $\delta / R_i < 1$ (since $p_i + R_i z_i \in \mathbb{B}_{R_i}(p_i) \subseteq \mathcal{X}_i$).

Estimation bias

The difference between this estimator and the oracle framework we discussed above is twofold. First, each player's *realized* action is $\hat{x}_i = x_i + \delta w_i$, not x_i, so there is a disparity between the point at which payoffs are queried and the action profile where the oracle is called. Second, the resulting estimator (SPSA) is not unbiased, so the analysis of the previous section does not apply (since it was carried out under the assumption that $B_t = 0$). In particular, given the feasibility adjustment (4.24), the estimate (SPSA) with $\hat{x}$ given by (4.25) satisfies

$$\mathbb{E}[\hat{v}_i] = \nabla_i u_i^\delta(x_i^\delta; x_{-i}^\delta), \tag{4.26}$$

7 For simplicity, we assume throughout this section that $\|\cdot\|$ is the Euclidean norm.

Algorithm 4.1: Bandit dual averaging [player indices suppressed]

Require: step-size sequence γ_t; query radius sequence δ_t; ball $\mathbb{B}_R(p) \subseteq \mathcal{X}$
choose $Y_1 \in \mathcal{Y}$ # initialization
for $t = 1, 2, \ldots$ **do**
  set $X_t \leftarrow Q(Y_t)$ # set pivot
  draw z_t uniformly from $\mathbb{S}^n$ # perturbation direction
  set $W_t \leftarrow z_t - R^{-1}(X_t - p)$ # query direction
  play $\hat{X}_t \leftarrow X_t + \delta_t W_t$ # choose action
  receive $\hat{u}_t \leftarrow u(\hat{X}_t)$ # get payoff
  set $V_t \leftarrow (n/\delta_t)\hat{u}_t \cdot z_t$ # estimate gradient
end for

so there are *two* sources of systematic error: an $\mathcal{O}(\delta)$ perturbation in the function, and an $\mathcal{O}(\delta)$ perturbation of each player's pivot point from x_i to x_i^δ.

To capture both sources of bias and separate them from the random noise, we will write

$$\hat{v}_i = V_i(x) + U_i + b_i \tag{4.27}$$

where $U_i = \hat{v}_i - \mathbb{E}[\hat{v}_i]$ and $b_i = \nabla_i u_i^\delta(x^\delta) - \nabla_i u_i(x)$. We are thus led to the following manifestation of the bias-variance dilemma: the bias term b in (4.27) is $\mathcal{O}(\delta)$, but the second moment of the noise term U is $\mathcal{O}(1/\delta^2)$; as such, an increase in accuracy (small bias) would result in a commensurate loss of precision (large noise variance).

Dual averaging with bandit feedback

Now, combining the learning framework of the previous section with (SPSA), we obtain the *bandit dual averaging* (BDA) algorithm:

$$\begin{aligned} \hat{X}_t &= X_t + \delta_t W_t \\ Y_{t+1} &= Y_t + \gamma_t V_t \\ X_{t+1} &= Q(Y_{t+1}) \end{aligned} \tag{BDA}$$

In the above, the perturbations W_t and the SPSA estimates V_t are respectively given by

$$W_{i,t} = z_{i,t} - R_i^{-1}(X_{i,t} - p_i) \qquad V_{i,t} = (n_i/\delta_t) u_i(\hat{X}_t)\, z_{i,t} \tag{4.28}$$

with each $z_{i,t}$ drawn independently and uniformly across players at each stage t; see also Algorithm 4.1 for a pseudocode implementation and Fig. 4.2 for a schematic representation.

Our first result below shows that (BDA) converges to equilibrium in monotone games:

Convergence in strictly monotone games

Theorem 4.15 (Bravo et al., 2018)**.** *Let $\mathcal{G} \equiv \mathcal{G}(\mathcal{N}, \mathcal{X}, u)$ be a strictly monotone game, and suppose that* (BDA) *is run with variable step-size γ_t and query radius δ_t such that*

$$\lim_{t\to\infty} \gamma_t = \lim_{t\to\infty} \delta_t = 0, \quad \sum_{t=1}^{\infty} \gamma_t = \infty, \quad \sum_{t=1}^{\infty} \gamma_t \delta_t < \infty, \quad \textit{and} \quad \sum_{t=1}^{\infty} \frac{\gamma_t^2}{\delta_t^2} < \infty. \tag{4.29}$$

Then, the sequence of realized actions $\hat{X}_t$ converges to the (necessarily unique) Nash equilibrium of $\mathcal{G}$ with probability 1.

Parameter tuning

Even though the setting is different, the conditions (4.29) for the tuning of the algorithm's parameters are akin to those encountered in Kiefer–Wolfowitz stochastic approximation schemes and serve a similar purpose. First, the conditions $\lim_{t\to\infty} \gamma_t = 0$ and $\sum_{t=1}^{\infty} \gamma_t = \infty$ respectively mitigate the method's inherent randomness and ensure a running horizon of sufficient length. The requirement $\lim_{t\to\infty} \delta_t = 0$ is also straightforward to explain: as players accrue more information, they need to decrease the sampling bias in order to have any hope of converging. However, decreasing δ also increases the variance of the players' gradient estimates, which might grow to infinity as $\delta \to 0$. The

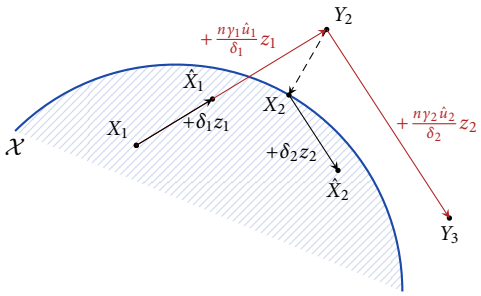


Figure 4.2: Schematic representation of (BDA) with (lazy) Euclidean projections. For clarity, we did not include the feasibility adjustment $R^{-1}(x-p)$ in the action selection step.

crucial observation here is that new gradients enter the algorithm with a weight of γ_t so the aggregate bias after t stages is of the order of $\mathcal{O}(\sum_{s=1}^t \gamma_s\delta_s)$ and its variance is $\mathcal{O}(\sum_{s=1}^t \gamma_s^2/\delta_s^2)$.

On the choice of γ_t and δ_t

Of course, since Theorem 4.15 is asymptotic in nature, it is not clear how to choose γ_t and δ_t so as to optimize the method's convergence rate. Heuristically, if we take schedules of the form $\gamma_t = \gamma/t^b$ and $\delta_t = \delta/t^c$ with $\gamma, \delta > 0$ and $0 < b, c \leq 1$, the only conditions imposed by (4.29) are $b + c > 1$ and $b - c > 1/2$. However, as we discussed above, the aggregate bias in the algorithm after t stages is $\mathcal{O}(\sum_{s=1}^t \gamma_t\delta_t) = \mathcal{O}(1/t^{b+c-1})$ and its variance is $\mathcal{O}(\sum_{s=1}^t \gamma_s^2/\delta_s^2) = \mathcal{O}(1/t^{2b-2c-1})$: if the conditions (4.29) are satisfied, both error terms vanish, but they might do so at very different rates. By equating these exponents in order to bridge this gap, we obtain $c = b/3$; moreover, since the single-shot SPSA estimator (SPSA) introduces a $\Theta(\delta_t)$ random perturbation, c should be taken as large as possible to ensure that this perturbation vanishes at the fastest possible rate. As a result, the most suitable choice for b and c seems to be $b = 1$, $c = 1/3$, leading to an error bound of $\mathcal{O}(1/t^{1/3})$.

We show below that this bound is indeed attainable for games that are *strongly monotone* in the sense of (2.77), i.e., there exists some $\alpha > 0$ such that

$$\langle V(x') - V(x), x' - x\rangle \leq -\alpha\|x' - x\|^2 \quad \text{for all } x, x' \in \mathcal{X}. \tag{4.30}$$

Focusing for expository reasons on the Euclidean incarnation of the method ($h(x) = \|x\|^2/2$ for $x \in \mathcal{X}$), we have:

Convergence rate in strongly monotone games

Theorem 4.16. *Let x^* be the (necessarily unique) Nash equilibrium of an α-strongly monotone game. If the players follow* (BDA) *with Euclidean projections and parameters $\gamma_t = \gamma/t$ and $\delta_t = \delta/t^{1/3}$ with $\gamma > 1/(3\alpha)$ and $\delta > 0$, we have*

$$\mathbb{E}[\|\hat{X}_t - x^*\|^2] = \mathcal{O}(t^{-1/3}). \tag{4.31}$$

Comparison with existing rates

We note in the above that the step-size schedule $\gamma_t \propto 1/t$ is not required in order to obtain an $\mathcal{O}(t^{-1/3})$ convergence rate: more general schedules of the form $\gamma_t \propto 1/t^b$ and $\delta_t \propto 1/t^c$ with $b > 3/4$ and $c = b/3 > 1/4$, still guarantee an $\mathcal{O}(t^{-1/3})$ rate of convergence for (BDA). To put things in perspective, if (DA) is run with first-order oracle feedback satisfying the statistical assumptions (2.24), the rate of convergence becomes $\mathcal{O}(1/t)$. Viewed in this light, the price for not having access to gradient information is no higher than $\mathcal{O}(t^{-2/3})$ in terms of the players' equilibration rate.

Finally, it is also worth comparing the bound (6.9) to the attainable rates for stochastic convex optimization (the single-player case). For problems with objectives that are both strongly convex and smooth, Agarwal et al. [2] attained an $\mathcal{O}(t^{-1/2})$ convergence rate with bandit feedback, which Shamir [138] showed is unimprovable. Thus, in the

Algorithm 4.2: EXP3 [player indices suppressed]

Require: step-size $\gamma_t > 0$; exploration factor $\delta_t > 0$
choose $Y_1 \in \mathcal{Y}$ # initialization
for $t = 1, 2, \ldots$ **do**
  set $X_t \leftarrow (1 - \delta_t)\,\Lambda(Y_t) + \delta_t \,\mathrm{unif}$ # choose mixed strategy
  play $\hat{a}_t \sim X_t$ # choose action
  receive $\hat{u}_t = u(\hat{a}_t)$ # get payoff
  set $V_t = (\hat{u}_t / X_{\hat{a}_t,t}) e_{\hat{a}_t}$ # estimate payoffs
  update $Y_{t+1} \leftarrow Y_t + \gamma_t V_t$ # update scores
end for

single-player case, the bound (6.9) is off by $t^{1/6}$ and coincides with the bound of Agarwal et al. [2] for strongly convex functions that are not necessarily smooth.

Averaging vs. last iterate

One reason for this gap is that the $\Theta(t^{-1/2})$ bound of Shamir [138] concerns the smoothed-out time average $\bar{X}_t = t^{-1} \sum_{s=1}^{t} X_s$, while our analysis concerns the sequence of *realized actions* $\hat{X}_t$. This difference is semantically significant: In optimization, the query sequence is just a means to an end, and only the algorithm's output matters (i.e., $\bar{X}_t$). In a game-theoretic setting however, it is the players' *realized* actions that determine their rewards at each stage, so the figure of merit is the actual sequence of play $\hat{X}_t$. This sequence is more difficult to control, so this disparity is, perhaps, not too surprising; nevertheless, we believe that this gap can be closed by using a more sophisticated single-shot estimate, e.g., as in the recent work of Bubeck and Eldan [31, 32]. We defer this analysis to future work.

4.3.2 *Payoff-based learning in finite games*

In the context of finite games, learning with bandit feedback is tantamount to players observing their realized in-game payoffs

$$\hat{u}_i = u_i(\hat{a}_i; \hat{a}_{-i}) \tag{4.32}$$

where $\hat{a}_i \in \mathcal{A}_i$ denotes the action chosen by the i-th player according to some mixed strategy $x_i \in \mathcal{X}_i \in \Delta(\mathcal{A}_i)$. Already, this shows that the finite game framework is markedly different from the continuous game framework studied in the previous section: the expected payoff $u_i(\hat{x})$ at a perturbed mixed strategy $\hat{x}$ cannot be observed, so the SPSA estimator used to run (BDA) cannot be employed either.

Importance sampling

The reason for this is that players have no information about the payoffs of strategies that were not chosen, so a new estimator must be constructed for that purpose. A standard way to do so is via the *importance sampling* (IS) estimator:

$$\hat{v}_{ia_i} = \frac{\mathbb{1}(a_i = \hat{a}_i)}{x_{ia_i}} \hat{u}_i = \begin{cases} \dfrac{u_i(\hat{a}_i; \hat{a}_{-i})}{x_{ia_i}} & \text{if } a_i = \hat{a}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.33}$$

Indeed, a straightforward calculation shows that

$$\begin{aligned} \mathbb{E}[\hat{v}_{ia_i}] &= \sum_{a_{-i} \in \mathcal{A}_{-i}} x_{-i,a_{-i}} \sum_{a_i' \in \mathcal{A}_i} x_{ia_i'} \frac{\mathbb{1}(a_i' = \hat{a}_i)}{x_{ia_i'}} u_i(a_i'; a_{-i}) \\ &= u_i(a_i; x_{-i}) \equiv u_{ia_i}(x). \end{aligned} \tag{4.34}$$

i.e., the estimator (4.33) is unbiased in the sense of (2.24). On the other hand, a similar calculation shows that the variance of $\hat{v}_{ia_i}$ grows as $\mathcal{O}(1/x_{ia_i})$, implying that (2.24c) may fail to hold if the players' action choice probabilities become arbitrarily small.

Motivated by the seminal work of [7], we will focus on a variant of (Hedge) with an explicit exploration factor which mixes the logit choice model with uniform action selection. The resulting algorithm is known as *exploration and exploitation with exponential weights* (EXP3), and it can be stated in recursive form as:

The EXP3 algorithm

$$\begin{aligned} X_{i,t} &= (1-\delta_t)\,\Lambda_i(Y_{i,t}) + \delta_t\,\mathrm{unif}_i \\ Y_{t+1} &= Y_{i,t} + \gamma_t V_{i,t} \end{aligned} \tag{EXP3}$$

where

1. $\delta_t > 0$ is a time-dependent exploration factor (discussed in detail below).
2. $\Lambda_i\colon \mathbb{R}^{\mathcal{A}_i} \to \Delta(\mathcal{A}_i)$ is the logit choice map (2.62).
3. $\mathrm{unif}_i = \mathbf{1}/|\mathcal{A}_i|$ denotes the uniform distribution on $\mathcal{A}_i$,
4. V_t is given by the estimator (4.33), viz.,

$$V_{ia_i,t} = \frac{\mathbb{1}(a_i = \hat{a}_{i,t})}{X_{ia_i,t}} u_i(\hat{a}_{i,t}; \hat{a}_{-i,t}), \tag{4.35}$$

   where $\hat{a}_{i,t} \in \mathcal{A}_i$ denotes the realized action of player i at time t.

For a pseudocode implementation, see also Algorithm 4.2 above.

To examine the equilibrium convergence properties of (EXP3), we will focus on the class of *potential games*, i.e., games that admit a *potential function* $F\colon \mathcal{A} \to \mathbb{R}$ such that

$$u_i(a_i'; a_{-i}) - u_i(a_i; a_{-i}) = F(a_i'; a_{-i}) - F(a_i; a_{-i}) \tag{4.36}$$

for all $a_i, a_i' \in \mathcal{A}_i$, $a_{-i} \in \mathcal{A}_{-i}$, and all $i \in \mathcal{N}$. This class of games is equivalent to the class of atomic non-splittable congestion games [124, 128] and has wide applications in operations research, economics, network design, and many other fields. As opposed to arbitrary finite games, potential games always admit pure Nash equilibria; moreover, such equilibria are generically strict.

In this context, we have the following general result for (EXP3):

Convergence of EXP3 in potential games

Theorem 4.17 (Cohen et al., 2017). *Let $\Gamma \equiv \Gamma(\mathcal{N}, \mathcal{A}, u)$ be a generic potential game. Suppose further that* (EXP3) *is run with a step-size sequence of the form $\gamma_t \propto 1/t^b$, $b \in (1/2, 1]$, and a decreasing exploration factor $\delta_t \downarrow 0$ such that*

$$\lim_{t\to\infty} \frac{\gamma_t}{\delta_t^2} = 0, \quad \sum_{t=1}^{\infty} \frac{\gamma_t^2}{\delta_t} < \infty, \quad \textit{and} \quad \lim_{t\to\infty} \frac{\delta_t - \delta_{t+1}}{\gamma_t^2} = 0. \tag{4.37}$$

Then, with probability 1, *X_t converges to a strict Nash equilibrium of* Γ.

The main challenge in proving Theorem 4.17 is that, unless the "innovation term" $U_t = V_t - V(X_t)$ has bounded variance, the general theory of Benaïm [14] does not imply that X_t forms an asymptotic pseudotrajectory of the underlying mean dynamics – here, the unperturbed replicator system (RD). Nevertheless, under the summability condition (4.37), it is possible to show that this is the case by using a martingale limit argument based on Burkholder's inequality. Furthermore, under the stated conditions, it is also possible to show that if X_t converges, its limit is necessarily a strict equilibrium of Γ.

Importantly, the summability condition (4.37) imposes a lower bound on the step-size exponent b: In particular, if $b = 1/2$, (4.37) cannot hold for any vanishing sequence of exploration factors $\delta_t \downarrow 0$. Given that the innovation term Z_t is bounded, we conjecture that this sufficient condition is not tight and can be relaxed further. This issue is left for future work.

We close this section by noting that Theorem 4.17 should be contrasted to earlier results by Kleinberg et al. [78] who showed that, after a transient stage of polynomial length, players end up playing a pure equilibrium for a fraction of time that is arbitrarily close to 1 with probability also arbitrarily close to 1. Mehta et al. [89] obtained a stronger result for (generic) 2-player coordination games, showing that the multiplicative weights algorithm (a linearized variant of the EW algorithm) converges to a pure Nash equilibrium for all but a measure 0 of initial conditions. However, in both these works, players are assumed to have full (though possibly imperfect) knowledge of their payoff vectors, including actions that were not chosen.

In a bona fide bandit framework, the closest antecedent to Theorem 4.17 is the work of Coucheney et al. [42] and Leslie and Collins [84] who showed that a "penalty-regulated" variant of (Hedge) converges to ε-logit equilibria in congestion games. In this light, Theorem 4.17 should also be contrasted to the results of Cominetti et al. [41] and Bravo [24] who established the convergence of an algorithm similar to (EXP3) in *any* game. The limit point of the algorithms considered by Cominetti et al. [41] and Bravo [24] is a logit equilibrium of the underlying game, though not necessarily an ε-equilibrium for arbitrarily small ε. Extending the analysis of Cominetti et al. [41] and Bravo [24] to the study of (EXP3) in arbitrary games is a very fruitful direction for future research.

Part II

APPLICATIONS

5

DISTRIBUTED OPTIMIZATION IN MULTIPLE-WORKER SYSTEMS

Very large-scale (VLS) optimization problems are often solved by distributing them over computer clusters and parallel computing grids capable of performing between 10^{15} and 10^{18} floating-point operations per second (in the exaFLOPS regime). However, this massive parallelization comes with its own unique set of challenges: independently coordinated computations and verification, fault detection and management, performance irregularities, and massive communication overhead are only some of the problems that arise in high-performance computing (HPC) applications. In this context, modeling each node of a computing cluster as an independent agent can provide valuable insights into optimizing the system's overall performance. Accordingly, the online learning methodologies and techniques developed in the previous chapters arise as a natural framework for examining large-scale distributed optimization problems.

DASGD schemes

One of the most widely used methods for solving VLS problems is *distributed asynchronous stochastic gradient descent* (DASGD), a family of algorithms that results from parallelizing stochastic gradient descent (SGD) on distributed computing architectures. However, a key obstacle in the efficient implementation of DASGD is the issue of *delays:* when a node contributes a gradient update in an online manner, the global model parameter may have already been updated by other nodes several times over, thereby rendering this gradient update stale. These delays can quickly add up if the computational throughput of a node is saturated, so the convergence of DASGD methods may be compromised in the presence of large delays. In the sections that follow, we use the online learning methodologies developed in the previous chapters to establish the convergence of DASGD methods, even when the observed delays grow large.

5.1 MULTIPLE-WORKER SYSTEMS

 # The following sections summarize results from [154, 155]

Distributed and stochastic optimization

Let $\mathcal{X}$ be a closed convex subset of $\mathbb{R}^n$. Throughout the sequel, we will focus on the stochastic optimization problem (Opt-S) first introduced in Chapter 2, viz.

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \mathcal{X}, \end{aligned} \tag{Opt-S}$$

where the objective function $f\colon \mathcal{X} \to \mathbb{R}$ is of the form

$$f(x) = \mathbb{E}[F(x;\omega)] \tag{5.1}$$

for some random function $F\colon \mathcal{X} \times \Omega \to \mathbb{R}$ defined over some (complete) probability space $(\Omega, \mathcal{F}, \mathbb{P})$. As is well known in the literature [18], the stochastic expectation in (5.1) contains as a special case the standard distributed optimization objective

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) \tag{5.2}$$

Algorithm 5.1: Master-slave implementation of stochastic gradient descent

Require: Master and K workers, $k = 1, \ldots, K$
1: **repeat**
2: **Master**:
(a) Pull stochastic gradient from worker
(b) Update current state
(c) Push state to worker
3: **Workers**:
(a) Pull state from master
(b) Compute stochastic gradient
(c) Push gradient update to master
4: **until** end

where each $f_i : \mathcal{X} \to \mathbb{R}$ is the loss associated with the i-th training sample.[1] Specifically, the general functional form (5.2) encompasses a wide variety of machine learning tasks ranging from empirical risk minimization with uniform weights to least squares and logistic regression, suppport vector machines (SVMs), matrix completion, and many other model-based problems.

Blanket assumptions

In terms of regularity, we make the following blanket assumptions in the sequel:

Assumption 5.1. The solution set $\mathcal{X}^* \equiv \arg\min f$ of (Opt-S) is nonempty.

Assumption 5.2. F is twice continuously differentiable in x for $\mathbb{P}$-almost all $\omega \in \Omega$.

Assumption 5.3. ∇F has bounded second moments, i.e., $\sup_{x \in \mathcal{X}} \mathbb{E}[\|\nabla F(x;\omega)\|_*^2] < \infty$.

By the dominated convergence theorem, Assumptions 5.2 and 5.3 together imply that f is differentiable and $\nabla f(x) = \nabla \mathbb{E}[F(x;\omega)] = \mathbb{E}[\nabla F(x;\omega)]$.[2] Assumption 5.2 then implies that ∇f is Lipschitz continuous. Since f is continuous and $\mathcal{X}$ is closed, the solution set $\mathcal{X}^*$ of (Opt-S) is itself closed. We will make free use of these facts in the sequel.

5.1.1 *Master-slave architectures*

Master-slave architectures

Our main goal here is to solve the optimization problem (Opt-S) in multiple-worker architectures, a widely used distributed computing framework for data-centers and parallel computing grids. One of the standard ways of deploying SGD methods in such systems – and that which we adopt in this section – is for the workers to asychronously compute stochastic gradients and then send them to the master,[3] while the master updates the global state of the system and pushes the update back to the workers [18, 145]. For a pseudocode implementation of this process, see Algorithm 5.1.

Due to the distributed nature of the master-slave system, a gradient received by the master on any given iteration can be stale: namely, there are delays in receiving local gradients from workers. As a simple example, consider a fully coordinated update scheme where each worker sends the computed gradient to and receives the updated iterate from the master following a round-robin schedule. In this case, each worker's gradient is received with a delay exactly equal to $K - 1$ (K is the number of workers in the system), because by the time the master receives worker K's computed gradient, the master has already applied $K - 1$ gradient updates from workers 1 to $K - 1$ (and since the schedule is round-robin, this delay of $K - 1$ is true for any one of the K workers).

1 For instance, this setup corresponds to empirical risk minimization with uniform weights.

2 Note that finite second moments automatically imply finite first moments, which in turns guarantees that the expectation of the gradient exists.

3 In machine learning applications, this is done by sampling a subset of the training data, computing the gradient for each data point and averaging over all points in the sample.

However, delays can be much worse since we allow full asynchrony: workers can compute and send (stochastic) gradients to the master without any coordinated schedule. In the asynchronous setting, fast workers (i.e., workers that are fast in computing gradients) will cause disproprotionately large delays to gradients produced by slow workers. Specifically, when a slower worker has finished computing a gradient, a fast worker may have already computed and communicated many gradients to the master. Since the master updates the global state of the system, one can gain a clearer representation of this scheme by looking at the master's update; we do so in Section 5.1.3 below.

5.1.2 *Multi-core systems with shared memory*

Multi-core systems

Another popular way of solving very large scale problems of the form (Opt-S) is by distributing them over multi-core clusters with shared memory. In this architecture, all processors can access a global memory which holds all the data needed for computing a gradient (as well as the system's state). The standard way of deploying SGD in such systems is for each processor to independently and asychronously read the current global iterate, compute a stochastic gradient, and then update the global iterate in the shared memory [18].[4] This process is presented below as Algorithm 5.2:

Algorithm 5.2: Multi-core stochastic gradient descent with shared memory

```
Require: K cores with shared memory
1: Commit initial state to memory
2: repeat
3:    do in parallel for each core
       (a) Pull current state
       (b) Compute stochastic gradient
       (c) Push updated state
4: until end
```

The key difference from Algorithm 5.1 is that there is no central entity that updates the global state; instead, each processor can both read the global state and update it. Since each core is performing the operations asynchronously, different cores may be reading the same global iterate at the same time. Further, the delays in this case are again caused by the heterogeneity across different cores: if a processor is slow in computing gradients, then by the time it finishes computing its gradient, the global state has been updated by other, faster processors many times over, thereby making its own gradient stale.

Here, we also adopt a common assumption that updating the global state is an atomic operation (and hence no two processors will be updating the global iterate at the same time). This is particularly true if the number of variables (i.e. the dimension of the decision variable) is not of a super-large scale (in the order of trillions of variables). The analysis presented here can be further extended to cases where only one variable or a small block of variables are being updated at a time. However, we omit this discussion because the resulting notation is quite onerous, and will obscure the main ideas behind an already complex framework.

5.1.3 *DASGD: A unified algorithmic representation*

The DASGD method

We now present a unified algorithmic description, aptly called *distributed asynchronous stochastic gradient descent* (DASGD), that formally captures both Algorithms 5.1 and 5.2. This process is encoded in pseudocode form as Algorithm 5.3 below.

4 This is again done by sampling a subset of the training data in the global memory and computing the gradient at the iterate for each datapoint and averaging over all the comptued gradients in the sample.

Algorithm 5.3: Distributed asynchronous stochastic gradient descent

Require: step-size sequence $\gamma_t > 0$
1: choose $Y_1 \in \mathbb{R}^n$ # initialization
2: **for** $t = 1, 2, \ldots$ **do**
3: set $X_t \leftarrow \Pi(Y_t)$ # state update
4: receive $V_t = -\nabla F(X_{s_t}; \omega_{s_t})$ # gradient update
5: set $Y_{t+1} = Y_t + \gamma_t V_t$ # gradient step
6: **end for**
7: **return** solution candidate X_t

In Algorithm 5.3, $t = 1, 2, \ldots$ is a global counter which is incremented every time an update occurs to the current solution candidate X_t (the global state): in master-slave systems, X_t is updated by the master; in multi-core systems, X_t is updated by each processor separately.

Since there are delays in both systems, the gradient applied to the current iterate X_t can be a gradient associated with a previous time step. This fact is abstractly captured by the second line in Algorithm 5.3. In full generality, we will write s_t for the iteration from which the gradient received at time t originated. In other words, the delay associated with iteration s_t is $t - s_t$, since it took $t - s_t$ iterations for the gradient computed on iteration s_t to be received at stage t. Note that s_t is always no larger than t; and if $t = s_t$, then there is no delay in iteration t.

Difference between architectures

The basic difference between the two distributed computing archecitures is reflected in the assumptions for s_t. Specifically, in master-slave systems, each $s(\cdot)$ is a one-to-one function because no two workers will ever receive the same update from the master (except possibly at initialization). On the other hand, in multi-core systems, s_t can be the same for different t's (since different processors may read the current iterate at the same time); however, it is easy to observe that the same s will appear at most K times for different t's, since there are K processors in total. Our analysis is *agnostic* to whether s_t is one-to-one or not so, in analysing the meta-algorithm DASGD, we obtain guarantees for both architectures simultaneously.

Notation-wise, we will write d_t for the delay required to compute a gradient requested at iteration t. This gradient is received at iteration $t + d_t$. Following this notation, the delay for a gradient received at t is $d_{s_t} = t - s_t$. Note also we have chosen the subscript associated with ω to be t: we can do so because ω_t's are i.i.d. (so the indexing is irrelevant).

5.2 ANALYSIS AND RESULTS

5.2.1 *Nonconvex unconstrained problems*

Motivated by its applications to machine learning models and neural network training, we begin with the case where $\mathcal{X} = \mathbb{R}^n$ and f is (possibly) non-convex. In this setting (and in the absence of more rigid assumptions), a standard metric to determine the stability of the algorithm is to show that gradients vanish in the long run.[5]

Relating delays to step-sizes

Our goal in this section will be to establish a mean square performance guarantee of DASGD in the presence of delays. Our main assumption in this regard will be as follows:

Assumption 5.4. The step-size sequence γ_t of DASGD (Algorithm 5.3) is tuned relative to the delay process d_t as follows:

5 An alternative phrase that is commonly used is that the so-called "criticality gap" vanishes. This is also colloquially – but unfortunately – referred to as "convergence to a critical point" in much of the machine learning literture.

1. *For bounded delays*, i.e., $\sup_t d_t \leq D$ for some $D > 0$:

$$\sum_{t=1}^{\infty} \gamma_t^2 < \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \gamma_t = \infty \tag{5.3a}$$

2. *For sublinearly growing delays*, i.e., $d_t = \mathcal{O}(t^p)$ for some $p \in (0, 1)$:

$$\gamma_t \propto \frac{1}{t} \tag{5.3b}$$

3. *For linearly growing delays*, i.e., $d_t = \Theta(t)$

$$\gamma_t \propto \frac{1}{t \log t} \tag{5.3c}$$

4. *For polynomially growing delays*, i.e., $d_t = \mathcal{O}(t^q)$ for some $q \geq 1$:

$$\gamma_t \propto \frac{1}{t \log t \log \log t} \tag{5.3d}$$

Step-sizes as delay mitigators

Note that as delays get larger, Assumption 5.4 prescribes less aggressive step-size policies. This is to be expected because the larger the delays, the more "averaging" one needs perform in order to mitigate the staleness that is caused by the delays (and smaller step-sizes correspond to averaging over a longer horizon). This is one of the important insights gained by the analysis to follow.

Another thing worth pointing out is that Assumption 5.4 also highlights the quantitative relationship between the class of delays and the class of step-sizes. For instance, when the delays increase from a linear rate to a polynomial rate, only a factor of $1/\log\log t$ needs to be added (which is effectively a constant). From a practical standpoint, this means that a step-size on the order of $1/(t \log t)$ will be a good model-agnostic choice and should suffice for almost all delay processes.

Convergence in non-convex problems

We now proceed to establish the theoretical convergence guarantees of DASGD for general non-convex objectives. By leveraging the Lipschitz continuity of the gradient, a telescoping sum argument, and a careful analysis of the interplay between delays and step-sizes, we obtain the following convergence result:

Theorem 5.1 (Zhou et al., 2018)**.** *Let X_t be the sequence of states generated by the DASGD algorithm* (*Algorithm 5.3*)*. Then, under Assumptions 5.1–5.4, we have:*

$$\lim_{t \to \infty} \mathbb{E}[\|\nabla f(X_t)\|_*^2] = 0. \tag{5.4}$$

Theorem 5.1 provides a fairly strong characterization of the long-run behavior of DASGD. In particular, it implies that the norm of the gradient vanishes in expectation, and that the gradient converges to 0 with high probability. In fact, if we strengthen Assumption 5.3 to posit that the problem's stochastic gradients are bounded almost surely (as opoosed to L^2), the statement of Theorem 5.1 can be likewised strengthened to almost sure convergence of $\|\nabla f(X_t)\|_*$ to 0. Given that stochastic gradients are uniformly bounded in model-based machine learning problems, this remark is particularly important for applications to machine learning and artificial intelligence.

5.2.2 *Convex problems*

We now turn to more structured problems and, in particular, convex ones:

Assumption 5.5. $f\colon \mathcal{X} \to \mathbb{R}$ is convex.

Of course, in contrast to the non-convex regime, convexity allows for a significantly finer convergence analysis and, in particular, targeting the algorithm's convergence to a global solution of (Opt-S). Our first result is an auxiliary proposition that is of independent interest: with probability 1, the sequence of states X_t generated by DASGD admits a subsequence converging to a global minimizer of (Opt-S). Formally, we have:

A convergent subsequence

Proposition 5.2. *Let X_t be the sequence of states generated by the DASGD algorithm (Algorithm 5.3). Then, under Assumptions 5.1–5.5, there exists with probability 1 a (possibly random) subsequence X_{t_k} of X_t such that $\lim_{k\to\infty} X_{t_k} \in \mathcal{X}^*$.*

The convexity of f (cf. Assumption 5.5 above) plays a crucial role in the proof of Proposition 5.2; for general nonconvex objectives, the arguments used in the proof of Proposition 5.2 do not suffice to establish subsequential convergence even to a critical point of f (at least, not without some extra structural assumption). On the other hand, the second important element of our analysis is not related to convexity; it states that X_t is an asymptotic pseudotrajectory of the continuous-time dynamics (CDA) with (lazy) Euclidean projections:

DASGD is an APT of (CDA)

Proposition 5.3. *Let X_t be the sequence of states generated by the DASGD algorithm (Algorithm 5.3). Then, under Assumptions 5.1–5.4, X_t is an APT of the dynamics* (CDA) *with $V(x) = -\nabla f(x)$.*

The convergence of the continuous-time dynamcis (CDA) to x^* essentially follows from the analysis of Chapter 3. However, this is not sufficient in itself to establish the convergence of X_t to a solution of (Opt-S). To do that, we further need to show with a separate energy argument that, if X_t gets ε-close to a minimizer of f (and, in particular, the subsequence limit point whose existence is guaranteed by Proposition 5.2), then it is "trapped" in a $\mathcal{O}(\varepsilon)$ neighborhood of said point for large enough t. This is done by controlling the "energy" (as defined by the Fenchel coupling) of the ODE trajectory, and then bounding its difference from the discrete-time sequence X_t via the APT property. Fleshing out this analysis, we finally obtain the following almost sure convergence result:

Global convergence in convex problems

Theorem 5.4. *Let X_t be the sequence of states generated by the DASGD algorithm (Algorithm 5.3) under Assumptions 5.1–5.5. Then, X_t converges to a (possibly random) solution of* (Opt-S) *with probability 1.*

We examine the practical implications of Theorems 5.1 and 5.4 in the next section.

5.2.3 *Numerical experiments*

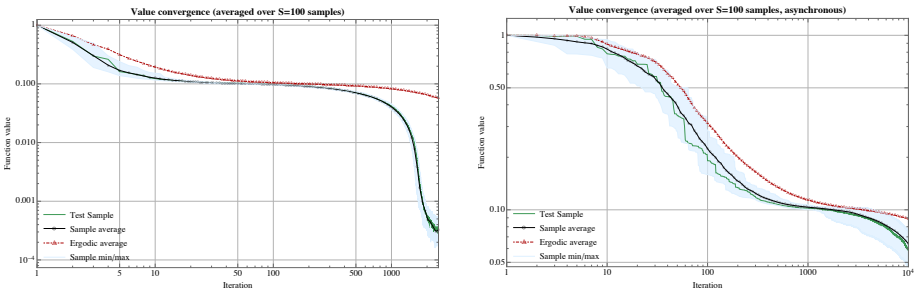
Performance in a nonconvex benchmark

We close our discussion with a suite of numerical experiments for DASGD. Specifically, we test the convergence of Algorithm 5.3 against a test function with $n = 10^5$ degrees of freedom based on the Rosenbrock optimization benchmark. Specifically, we consider the objective

$$f(x) = \sum_{i=1}^{10^5-1} [10^5(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \tag{5.5}$$

with $x_i \in [0, 2]$, $i = 1, \dots, 10^5$. The global minimum of f is located at $(1, \dots, 1)$, at the end of a very thin and very flat parabolic valley which is notoriously difficult for first-order methods to traverse [123]. Since the minimum of the Rosenbrock function is known, it is easy to validate the performance of DASGD methods in this setting.

For our numerical experiments, we considered *a*) a synchronous update schedule as a baseline; and *b*) an asynchronous master-slave framework with random delays that scale as $d_t = \Theta(t)$. In both cases, Algorithm 5.3 was run with a decreasing step-size of the form $\gamma_t \propto 1/(t \log t)$ and stochastic gradients drawn from a standard multivariate Gaussian distribution (i.e., zero mean and identity covariance matrix).



(a) Convergence with no delays

(b) Convergence under linearly growing delays

Figure 5.1: Convergence of DASGD in a non-convex problem with $n = 10^5$ degrees of freedom.

Our results are shown in Fig. 5.1. Starting from a random (but otherwise fixed) initial condition, we ran $S = 10^5$ realizations of DASGD (with and without delays). We then plotted a randomly chosen trajectory ("test sample" in Fig. 5.1), the sample average, and the min/max over all samples at every update epoch. For comparison purposes, we also plotted the value of the so-called "ergodic average"

$$\bar{X}_t = \frac{\sum_{s=1}^{t} \gamma_s X_s}{\sum_{s=1}^{t} \gamma_s}, \tag{5.6}$$

which is often used in the analysis of DASGD in the convex case.

Even though this averaging leads to very robust convergence rate estimates in the convex case, we see here that it performs worse than the worst realization of DASGD. The reason for this is the lack of convexity: due to the ridges and talwegs of the Rosenbrock function, Jensen's inequality fails dramatically to produce an improvement over X_t (and, in fact, causes delays as it causes X_t to deviate from its gradient path). Consequently, this simple suite of experiments indicates that establishing convergence of the iterate X_t itself is not only theoretically stronger than convergence of the ergodic average, but also leads to better results in non-convex problems.

6

SIGNAL COVARIANCE OPTIMIZATION IN WIRELESS NETWORKS

In this chapter, we apply the online learning techniques discussed earlier in this manuscript to derive and analyze the *matrix exponential learning* (MXL) algorithm, a semidefinite optimization method for throughput maximization in multiple-input and multiple-output (MIMO) systems – also known as Gaussian vector channels in signal processing and information theory. After introducing the problem in Section 6.1 below, we present the MXL algorithm and its main performance guarantees in Section 6.2, and we provide a set of experiments under realistic channel conditions in Section 6.3.

6.1 SYSTEM MODEL AND ASSUMPTIONS

 # The following sections summarize results from [91, 98, 100]

MIMO channel model

A MIMO multiple access channel (MAC) consists of a finite set of wireless devices $k \in \mathcal{K} \equiv \{1, \dots, K\}$ that transmit simultaneously over a common channel to a base receiver with N antennas. If the k-th transmitter is equipped with M_k transmit antennas, the signal at the receiver can be expressed via the standard baseband model

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{x}_k + \mathbf{z}, \tag{6.1}$$

where:

1. $\mathbf{x}_k \in \mathbb{C}^{M_k}$ is the signal transmitted by the k-th device.
2. $\mathbf{y} \in \mathbb{C}^N$ denotes the aggregate signal at the receiver.
3. $\mathbf{H}_k \in \mathbb{C}^{N \times M_k}$ is the $N \times M_k$ channel matrix of the k-th device.
4. $\mathbf{z} \in \mathbb{C}^N$ is the ambient noise in the channel, including thermal, atmospheric and other peripheral interference effects (and modeled for simplicity as a zero-mean, circulant Gaussian vector with unit covariance).

In this context, the *transmit power* of the k-th device is simply

$$p_k = \mathbb{E}\left[\|\mathbf{x}_k\|^2\right] = \operatorname{tr}(\mathbf{Q}_k), \tag{6.2}$$

where $\mathbf{Q}_k$ denotes the corresponding *signal covariance matrix*

$$\mathbf{Q}_k = \mathbb{E}\left[\mathbf{x}_k \mathbf{x}_k^\dagger\right] \tag{6.3}$$

and the expectation is taken over the Gaussian codebook of the k-th device. Hence, assuming that each device's maximum transmit power is finite, we obtain the feasibility constraints:

$$\mathbf{Q}_k \succcurlyeq 0 \quad \text{and} \quad \operatorname{tr}(\mathbf{Q}_k) \leq P_k, \tag{6.4}$$

where $P_k > 0$ denotes the maximum transmit power of the k-th device.

Shannon–Telatar capacity

Our analysis focuses on *static channels*, i.e., $\mathbf{H}_k$ will be assumed to remain constant (or nearly constant) throughout the transmission horizon. In this case, assuming that

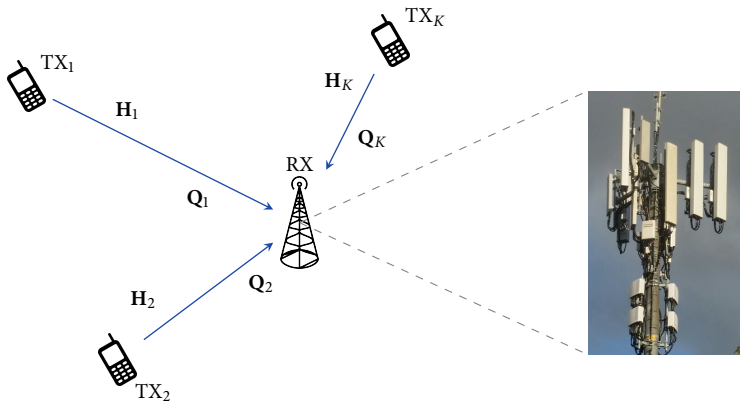


Figure 6.1: A MIMO multiple access channel network.

interference by all other devices is treated as additive noise at the receiver, the achievable transmission rate of each device is given by the Shannon-Telatar expression [144]:

$$u_k(\mathbf{Q}) = \log\det\left(\mathbf{I} + \textstyle\sum_\ell \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger\right) - \log\det\left(\mathbf{W}_{-k}\right), \tag{6.5}$$

where $\mathbf{Q} = (\mathbf{Q}_1, \ldots, \mathbf{Q}_K)$ and

$$\mathbf{W}_{-k} = \mathbf{I} + \textstyle\sum_{\ell \neq k} \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger \tag{6.6}$$

represents the multi-user interference (MUI) covariance matrix of the k-th device. We will thus say that a transmit profile $\mathbf{Q}^\star = (\mathbf{Q}_1^\star, \ldots, \mathbf{Q}_K^\star)$ is at *Nash equilibrium* when no device can unilaterally improve his individual achievable rate u_k, i.e.

$$u_k(\mathbf{Q}^\star) \geq u_k(\mathbf{Q}_k; \mathbf{Q}_{-k}^\star) \quad \text{for all } \mathbf{Q}_k \in \boldsymbol{\mathcal{Q}}_k,\ k \in \mathcal{K}, \tag{6.7}$$

where $(\mathbf{Q}_k; \mathbf{Q}_{-k}^\star)$ is shorthand for $(\mathbf{Q}_1^\star, \ldots, \mathbf{Q}_k, \ldots, \mathbf{Q}_K^\star)$ and

$$\boldsymbol{\mathcal{Q}}_k = \left\{\mathbf{Q}_k \in \mathbb{C}^{M_k \times M_k} : \mathbf{Q}_k \succcurlyeq 0, \mathrm{tr}(\mathbf{Q}_k) \leq P_k\right\} \tag{6.8}$$

denotes the set of feasible signal covariance matrices for the k-th device.

Throughput maximization

Dually to the above, if the receiver employs successive interference cancellation to decode the received messages, the network's achievable sum rate will be [152]:

$$R(\mathbf{Q}) = \log\det\left(\mathbf{I} + \textstyle\sum_k \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger\right). \tag{6.9}$$

In this way, we obtain the sum rate maximization problem:

$$\begin{array}{ll} \text{maximize} & R(\mathbf{Q}), \\ \text{subject to} & \mathbf{Q}_k \in \boldsymbol{\mathcal{Q}}_k,\ k = 1, \ldots, K. \end{array} \tag{RM}$$

As can be easily checked, the sum rate function (6.9) is a *potential function* for the game (6.5) in the sense that

$$u_k(\mathbf{Q}_k; \mathbf{Q}_{-k}) - u_k(\mathbf{Q}_k'; \mathbf{Q}_{-k}) = R(\mathbf{Q}_k; \mathbf{Q}_{-k}) - R(\mathbf{Q}_k'; \mathbf{Q}_{-k}). \tag{6.10}$$

Hence, with R concave, it follows that the solutions of the Nash equilibrium problem (6.7) coincide with the solutions of (RM); put differently, optimizing the network's achievable sum rate (6.9) under successive interference cancellation is equivalent to reaching a Nash equilibrium with respect to the users' individual achievable rates (6.5) under. For

concreteness, we will focus throughout on the sum rate maximization problem (RM); however, owing to the above observation, the equilibrium problem (6.7) can be handled in a similar manner.

6.2 MATRIX EXPONENTIAL LEARNING

Water-filling

The sum rate maximization problem (RM) is traditionally solved by water-filling (WF) methods [38], either iterative [133, 152] or simultaneous [132]. More precisely, transmitters are typically assumed to have perfect knowledge of the channel matrices $\mathbf{H}_k$ and the aggregate signal-plus-noise covariance matrix

$$\mathbf{W} = \mathbb{E}[\mathbf{y}\mathbf{y}^\dagger] = \mathbf{I} + \textstyle\sum_\ell \mathbf{H}_\ell \mathbf{Q}_\ell \mathbf{H}_\ell^\dagger, \tag{6.11}$$

which is in turn used to calculate the MUI covariance matrices $\mathbf{W}_{-k} = \mathbf{W} - \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger$ and "water-fill" the effective channel matrices $\widetilde{\mathbf{H}}_k = \mathbf{W}_{-k}^{-1/2} \mathbf{H}_k$ at the transmitter [152]. At a multi-user level, this water-filling process could take place either iteratively (with users updating their covariance matrices in a round robin fashion) [152] or simultaneously (with all users updating at once) [132]. The former (iterative) scheme converges always (but slowly for large numbers of users) [152], whereas the latter (simultaneous) algorithm is much faster [132] but it may fail to converge, even in simple, 2-user parallel multiple access channels [99].

6.2.1 *The matrix exponential learning algorithm*

Matrix exponential learning

Instead of relying on fixed-point methods, we will take an approach based on dual averaging: specifically, we will track the direction of steepest ascent of the system's sum rate in a dual, unconstrained space, and then map the result back to the problem's feasible space via matrix exponentiation. Formally, assuming for the moment perfect feedback, we will consider the matrix exponential learning scheme:

$$\begin{aligned} \mathbf{Y}_{k,t+1} &= \mathbf{Y}_{k,t} + \gamma_t \mathbf{V}_k(\mathbf{Q}_t), \\ \mathbf{Q}_{k,t+1} &= P_k \frac{\exp(\mathbf{Y}_{k,t+1})}{\operatorname{tr}[\exp(\mathbf{Y}_{k,t+1})]}, \end{aligned} \tag{MXL}$$

where:

1. $t = 1, 2, \dots$ denotes the algorithm's iteration counter.
2. $\mathbf{V}_k \equiv \mathbf{V}_k(\mathbf{Q})$ denotes the (matrix) derivative of the system's sum rate with respect to each user's covariance matrix, viz.
$$\mathbf{V}_k(\mathbf{Q}) \equiv \nabla_{\mathbf{Q}_k} R(\mathbf{Q}) = \nabla_{\mathbf{Q}_k} u_k(\mathbf{Q}) = \mathbf{H}_k^\dagger \mathbf{W}^{-1} \mathbf{H}_k. \tag{6.12}$$
3. $\mathbf{Y}_k$ is a gradient aggregation matrix with a role similar to (DA).[1]
4. γ_t is a decreasing step-size sequence.

Intuitively, (MXL) assigns more power to the spatial eigendirections that perform well while the variable step-size γ_t keeps the eigenvalues of $\mathbf{Q}_t$ from approaching zero too fast. Of course, to employ the recursion (MXL), each user $k \in \mathcal{K}$ needs to know their

1 Specifically, its role is to reinforce the spatial directions that lead to higher sum rates by increasing the corresponding eigenvalues of $\mathbf{Q}_k$.

individual gradient matrix $\mathbf{V}_k$. In turn, this matrix requires knowledge of $\mathbf{H}_k$ and the received signal precision matrix

$$\mathbf{P} = \mathbf{W}^{-1} = \left(\mathbf{I} + \sum_k \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^\dagger\right)^{-1}. \tag{6.13}$$

Gradient estimation

Of these two matrices ($\mathbf{H}_k$ and $\mathbf{P}$), the former can be estimated by pilot signals, and is assumed known at the receiver [72]. As for the latter, since the channel is assumed Gaussian, $\mathbf{P}$ can be estimated by means of the bias-adjusted estimator

$$\hat{\mathbf{P}} = \frac{S - N - 1}{S} \hat{\mathbf{W}}^{-1}, \tag{6.14}$$

where $\hat{\mathbf{W}} = S^{-1} \sum_{s=1}^{S} \mathbf{y}_s \mathbf{y}_s^\dagger$ is an (unbiased) estimate for the received signal covariance matrix $\mathbf{W}$ [5]. In more detail, if each transmitter takes S independent measurements $\hat{\mathbf{H}}_{k,1}, \dots, \hat{\mathbf{H}}_{k,S}$ of their channel matrix (e.g., via independent reverse pilot sampling), an unbiased estimate for $\mathbf{V}_k$ is given by the expression:

$$\hat{\mathbf{V}}_k = \frac{1}{S(S-1)} \sum\nolimits_{s \neq s'} \hat{\mathbf{H}}_{k,s}^\dagger \hat{\mathbf{P}} \hat{\mathbf{H}}_{k,s'}, \tag{6.15}$$

where $\hat{\mathbf{P}}$ is the latest estimate of (6.14) of $\mathbf{W}^{-1}$ that was broadcast by the receiver. Indeed, given that the sampled channel matrix measurements $\hat{\mathbf{H}}_{k,s}$ are assumed stochastically independent, we readily obtain:

$$\mathbb{E}[\hat{\mathbf{V}}_k] = \frac{1}{S(S-1)} \sum\nolimits_{s \neq s'} \mathbb{E}\left[\hat{\mathbf{H}}_{k,s}^\dagger \hat{\mathbf{P}} \hat{\mathbf{H}}_{k,s'}\right] = \mathbf{H}_k^\dagger \mathbf{W}^{-1} \mathbf{H}_k, \tag{6.16}$$

i.e., (6.15) constitutes an unbiased estimator of $\mathbf{V}$.

The construction above provides an estimator $\hat{\mathbf{V}}$ with $\mathbb{E}[\hat{\mathbf{V}}] = \mathbf{V}$. As for the variance of $\hat{\mathbf{V}}$, (6.15) can also be used to derive an expression for $\mathrm{Var}(\hat{\mathbf{V}})$ in terms of the moments of $\hat{\mathbf{P}}$ and $\hat{\mathbf{H}}$. Since the system input and noise are assumed Gaussian, the former are all finite (and Gaussian-distributed), implying in turn that $\hat{\mathbf{V}}$ satisfies the requirements (2.24) for an unbiased stochastic first-order oracle with uniformly bounded variance.

6.2.2 *Performance guarantees*

We are now in a position to state our main result for (MXL):

Convergence of MXL

Theorem 6.1 (Mertikopoulos and Moustakas, 2016)**.** *Assume that* (MXL) *is run with nonincreasing step sizes γ_t such that $\sum_t \gamma_t^2 < \sum_t \gamma_t = \infty$ and gradient feedback of the form* (6.15)*. Then, $\mathbf{Q}_t$ converges to the solution set $\mathcal{Q}$ of the sum rate maximization problem* (RM) *with probability* 1*.*

Moreover, if $\bar{\mathbf{Q}}_t = \sum_{s=1}^t \gamma_s \mathbf{Q}_s \big/ \sum_{s=1}^t \gamma_s$ denotes the ergodic average of $\mathbf{Q}_t$, we have:

$$\mathbb{E}[R(\bar{\mathbf{Q}}_t)] \leq R_{\max} - \varepsilon_t \tag{6.17}$$

and

$$\mathbb{P}(R_{\max} - R(\bar{\mathbf{Q}}_t) \geq \alpha) \leq \exp\left(-\frac{\theta_t^2 \alpha^2}{8K^2 \sum_{s=1}^t \gamma_s^2 \sigma_s^2}\right) \tag{6.18}$$

where

$$\varepsilon_t = \frac{\sum_{k=1}^K \log M_k + \frac{1}{2} L^2 \sum_{s=1}^t \gamma_s^2}{\sum_{s=1}^t \gamma_s}, \tag{6.19}$$

$\theta_t = \sum_{s=1}^t \gamma_s$, and L^2 is a positive constant depending only on the users' maximum powers and their channel gain matrices.

Per iteration complexity of MXL

In terms of per iteration complexity, we should note that each iteration of (MXL) is polynomial in the number of transmit and receive antennas (for calculations at the transmitter and receiver side respectively). Specifically, the complexity of the required matrix inversion and exponentiation steps is $\mathcal{O}(N^{\omega})$ and $\mathcal{O}(M_k^{\omega})$ respectively, where the exponent ω can be taken as low as 2.373 if the processing units employ fast Coppersmith–Winograd matrix multiplication methods [47].The Hermitian structure of $\mathbf{W}$ can be exploited to reduce the computational cost of each iteration even further but we do not address such issues: In practice, the number of transmit and receive antennas are physically constrained by the size of the wireless array, so these operations are quite light.

By comparison, the computational bottleneck of each iteration in distributed water-filling is the calculation of the effective channel matrix $\widetilde{\mathbf{H}}_k = \mathbf{W}_k^{-1/2}\mathbf{H}_k$ of each user and, subsequently, sorting the singular values of $\widetilde{\mathbf{H}}_k$. The computational complexity (per user) of these operations is $\mathcal{O}(\max\{M_k, N\}^{\omega})$ and $\mathcal{O}(M_k \log M_k)$ respectively, leading to an overall complexity of $\mathcal{O}(\max\{M_k, N\}^{\omega})$. This is the same complexity of water-filling methods, so (MXL) is no worse off in this regard either.

Outage probabilities

Finally, we should note that the bounds (6.18) represent the probability of observing sum rates far below the channel's capacity so they can be interpreted as a measure of the system's outage probability. In this context, the tail behavior of (6.18) shows that (MXL) hardens considerably around its deterministic limit: even though measurement errors can become arbitrarily large, the probability of observing sum rates much lower than what is obtainable with perfect gradient measurements decays very fast. In fact, this rate of decay is exponential: for large t, the factor $\theta_t^{-2}\sum_{s=1}^{t}\gamma_s^2$ which controls the width of non-negligible large deviations in (6.18) is of order $\mathcal{O}(1/n)$ for step-size sequences of the form $\gamma_t \propto 1/t^b$, $b \in (0, 1/2)$, and of order $\mathcal{O}(t^{2b-2})$ for $b \in (1/2, 1)$.

6.3 NUMERICAL EXPERIMENTS IN MIMO NETWORKS

Water-filling vs. MXL

To assess the performance of (MXL) in practical scenarios, we present below a series of numerical experiments. First, in Fig. 6.2, we investigate the convergence speed of (MXL) as a function of the number of wireless transmitters and transmit/receive antennas, using state-of-the-art water-filling (WF) methods as a benchmark. For concreteness, we compared the evolution of (MXL) to that of iterative and simultaneous water-filling for a system consisting of a base MIMO terminal with $N = 16$ receive antennas and $K = \{20, 50\}$ wireless users. We then plotted the users' Shannon rate (6.9) at each iteration; for comparison, we also plotted the channel's sum capacity and the users' sum rate under uniform power allocation.

As can be seen in Fig. 6.2, the MXL algorithm attains the system's sum capacity within a few iterations (essentially within a single iteration for $K = 50$ users).[2] This convergence behavior represents a marked improvement over traditional WF methods, even in moderately-sized systems with $K = 20$ users. First, iterative water-filling is much slower than (MXL); second, simultaneous water-filling may fail to converge altogether due to "ping-pong" effects that occur when the users change transmit eigenvalues at the same time. By contrast, (MXL) converges very quickly, even for large numbers of users and/or antennas per user.

Scalability and robustness

The scalability and robustness of (MXL) is further examined in Fig. 6.3 where we plot the number of iterations required for users to attain 99% of the system's sum capacity. More precisely, for each value of K and N in Fig. 6.3, we ran the MXL algorithm for 100 network instantiations (with simulation parameters as before) and we plotted the average number of iterations required to attain 99% of the network's capacity. This process was

2 Alternatively, in the game-theoretic context of (6.7), this implies that the system's users reach a unilaterally stable Nash equilibrium.

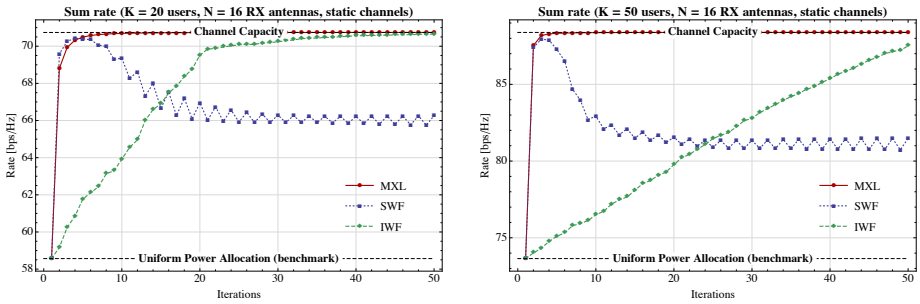
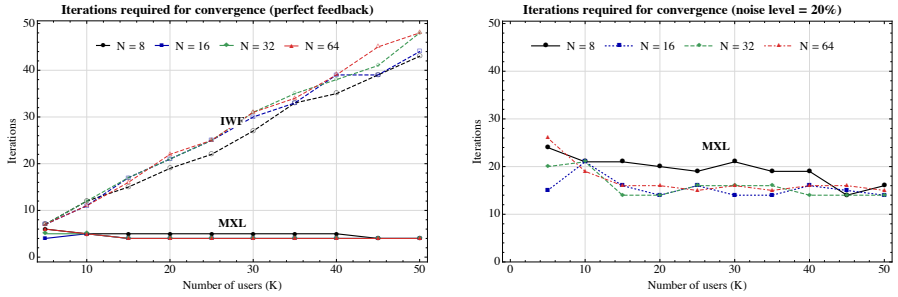


Figure 6.2: Comparison of matrix exponential learning (MXL) to water-filling (WF) methods. The iterative water-filling algorithm converges slowly because only one user updates per cycle; the simultaneous variant (SWF) is much faster (because all users updates simultaneously), but it may fail to converge due to the appearance of best-response cycles in the update process. By contrast, (MXL) converges within a few iterations, even for large K.



(a) Learning with perfect measurements

(b) Learning with imperfect measurements

Figure 6.3: Scalability of (MXL) under perfect and imperfect feedback (Figs. 6.3a and 6.3b respectively). The convergence threshold was set to 99% of the system's sum capacity and the number of iterations required for convergence was averaged over 100 realizations. In Fig. 6.3a, we also plotted the corresponding data for the iterative water-filling algorithm (dashed lines with open markers).

repeated for both perfect and imperfect feedback (with a 20% relative error level), and the results were plotted in Figures 6.3a and 6.3b respectively. For comparison purposes, we also plotted the number of iterations required for the convergence of iterative water-filling (IWF) in the case of perfect feedback; since simultaneous water-filling (SWF) often fails to converge, it was not included in our benchmark considerations (and likewise for IWF under imperfect feedback).

As can be seen in Fig. 6.3, (MXL) scales very well with the number of users (and/or antennas per user), achieving the system's sum capacity within (roughly) the same number of iterations. In fact, (MXL) is *faster* in larger systems because users can employ a more aggressive step-size policy.[3] Of course, in the case of imperfect feedback (Fig. 6.3b), users have to be less aggressive because erroneous observations can perturb the algorithm's performance. For this reason, (MXL) with imperfect feedback converges more slowly, but it still attains the system's sum capacity within roughly the same number of iterations, independently of the number of users and/or antennas per user in the system.

Per iteration complexity and wall-clock time

The (per user) computational cost of each iteration of (MXL) is examined in Fig. 6.4. Specifically, in Fig. 6.4, we focused on a system with $N \in [4, 64]$ receive antennas and $K = 50$ transmitters, each with a number of transmit antennas drawn randomly between 2 and $N/2$. We then plotted the actual CPU time required to perform one iteration of

3 In large systems, the optimal signal covariance profile $\mathbf{Q}^\star$ has many zero eigenvalues. As a result, using a very large step-size allows users to approach $\mathbf{Q}^\star$ within very few iterations, with no danger of oscillations.

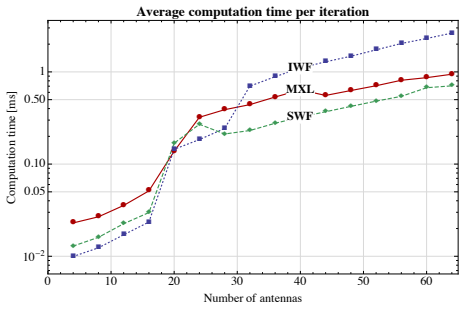


Figure 6.4: Average computation time per user and per iteration. Each iteration of (MXL) exhibits the same complexity behavior as water-filling methods.

(MXL) (per user) on a typical mid-range commercial laptop, averaging over 100 system realizations. For comparison, we also plotted the corresponding computation times for iterative and simultaneous water-filling (always per user and per iteration). As can be seen, the computational cost of (MXL) lies between that of IWF and SWF and is quite low, even for large number of antennas per user. Specifically, the computational time required to perform one iteration of (MXL) is well below the typical frame duration ($\delta = 5\,\text{ms}$), even for several tens of transmit/receive antennas.

7

PERSPECTIVES

A natural refinement of the questions treated so far would be to *a*) characterize the classes of games that are "learnable"; and *b*) to provide efficient learning algorithms that remain convergent in the broadest possible class of games (e.g., beyond monotone/coherent minds). This general framework opens up several questions for future research.

TOWARDS A HODGE THEORY OF GAMES. The Hodge decomposition theorem is a fundamental result in differential geometry which, in a greatly simplified form, states that any sufficiently smooth and rapidly decaying vector field can be decomposed as the sum of an irrotational (i.e., curl-free) and a solenoidal (i.e., divergence-free) vector field. This decomposition is of great importance because dynamical systems that are solenoidal are typically recurrent (i.e., they exhibit cycles) while dynamical systems that are irrotational are typically convergent. This dichotomy naturally invites a comparison with the behavior of online learning in games (convergence in "stable" games, and cycles in zero-sum games).

In a recent paper, Candogan et al. [34] derived a geometric categorization of finite games based on the Hodge decomposition theorem for graphs (but did not provide any insights about the behavior of learning based on this decomposition). The first question that arises in this context is whether this decomposition can be used to better understand evolutionary/learning dynamics in finite games; moreover, it is also natural to ask whether a Hodge-like decomposition can be found for general games with continuous action sets (which are of crucial importance in artificial intelligence and its applications).

Specifically, by mapping a game to a canonical 1-form which admits a Hodge decomposition into an exact (potential), a harmonic, and a co-exact component, concrete questions that arise are *a*) whether the decomposition of Candogan et al. [34] can provide the basis for a general Hodge theory of games; *b*) what is the role played by the choice of geometry (Riemannian metric) in determining the components of the Hodge decomposition; and *c*) whether the Hodge components are always consistent with a class of associated learning algorithhms. For instance, it is well known that the replicator dynamics can be seen as a form of Shahshahani gradient descent: could a Shahshahani–Hodge decomposition provide different insights for learning in games?

THE ROLE OF MEMORY IN GAME-THEORETIC LEARNING. Depending on how players aggregate past observations (i.e., whether they are treating them on an equal basis or if they discount past observations in favor of newer ones), the outcome of a learning process could vary dramatically. Quite surprisingly, preliminary results show that "nostalgic" players who assign more weight to past events may exhibit very strong rationality properties, such as the elimination of weakly dominated strategies (at least when the game does not change) [82]. Such phenomena are rather counter-intuitive, so an important open question is to develop a unified framework for the study of different valuations of past events, and to chart the properties of online learning in this context.

ACCELERATED LEARNING IN GAMES. One of the most widely lauded advances in optimization theory in the 80's was Nesterov's "fast gradient" algorithm – also known as *accelerated gradient descent*. This technique achieves the fastest possible convergence rate for convex problems with smooth objectives (i.e., with a Lipschitz gradient), but its performance in a game-theoretic setting is completely unknown. Motivated by recent results for extra-gradient techniques in games [102], a natural question that arises is the study of the no-regret and convergence properties of Nesterov's method in games and online variational inequalities, with a view towards providing faster regret minimization and equilibrium convergence rates in games.

MULTI-AGENT LEARNING IN NON-STATIONARY ENVIRONMENTS. Game-theoretic learning has focused almost exclusively on the case where the game itself is *static*: each player's individual payoff function might vary as a function of the other players' actions, but the mechanism underlying the players' interactions – i.e., the actual game – does not change with time. However, since real-world scenarios are rarely stationary, a key issue that arises is to truly mix the non-stationary framework of online optimization to the multi-agent setting of normal form games.

Specifically, focusing on games that evolve over time, if all players follow a learning policy that leads to no dynamic regret, does the induced sequence of play remain close to the Nash set of the game as it evolves over time? A promising entry point to this question is in the context of slowly-varying games (i.e., sequences of normal form games whose total variation grows sublinearly in time), games that admit a limit, and other relevant classes of time-varying games.

CONVERGENCE IN NON-MONOTONE SADDLE-POINT PROBLEMS. In a more practical setting, the adversarial match-up of deep learning mechanisms has led to extraordinary advances in the field of artificial intelligence, not the least of which is the ability to pass a specific version of Turing's test (the automatic generation of images that can fool a human observer). However, despite the highly promising results they provide, our theoretical understanding of GANs is still at an embryonic stage: the research community has a partial idea of "what" works in practice, but not the "why" or the "how".

Typically, deep learning involves non-convex loss functions for which finding even local minima is NP-hard; nevertheless, elementary techniques such as SGD (and other first-order methods) seem to work fairly well in practice. For this class of problems, recent results have started providing useful theoretical insights, but several key questions remain: Under which conditions is it reasonable to expect concrete convergence guarantees? Are the existing optimization algorithms guaranteed to avoid limit cycles and/or other spurious critical sets? If not, what should they be replaced with?

To provide concrete answers to the above questions, it would be interesting to employ methodologies and techniques from the theory of dynamical systems and differential geometry. Particularly promising would be the use of Morse theory and center manifold theorems to examine whether it is possible for a given algorithm to admit limit points – or, more generally, *limit sets* – that are not saddle points of the problem at hand. As such, a second direction to examine would be to study convergence under local versions of the variational coherence property of [102]: this would allow a better understanding of the successes and failures of first-order methods in adversarial learning models, and would provide a principled methodology for adversarial neural network training.

PARTICLE GANS. In tandem to the above, an open question is whether an evolutionary approach to GAN training (e.g., via particle swarm techniques) can bring practical benefits. Specifically, by initializing the training weights of a neural net at randomly selected points, it is possible to generate a "training swarm" that can be modeled as a

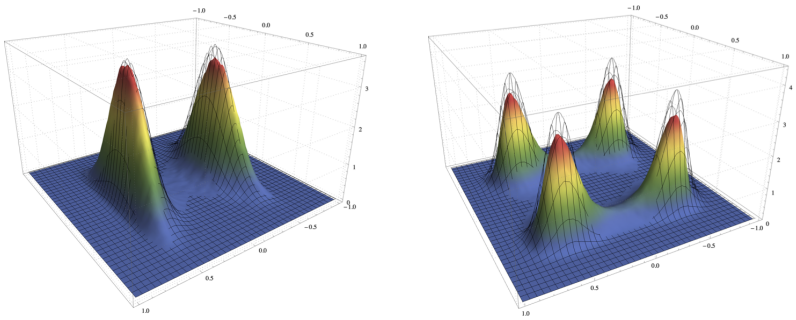


Figure 7.1: GAN training based on stochastic gradient descent in multi-modal saddle-point problems (left: a problem with two modes; right: a four-mode problem). Each surface plot was generated by drawing 1000 initializations of a GAN, and plotting a histogram of all points visited; the wireframe represents a theoretical estimate based on the Fokker-Planck equation, indicating a remarkably close agreement between theory and practice.

Langevin dynamical system (not unlike Einstein's original study of Brownian motion). The stationary state of this process (corresponding to the average "trained" network) can be obtained by solving the corresponding Fokker-Planck equation.

Of course, solving a Fokker-Planck equation is a task of considerable difficulty in itself; furthermore, despite the extensive literature surrounding the Fokker-Planck equation, very few works have treated the case where the underlying stochastic process does not admit a potential. Since GANs are *de facto* multi-agent problems (as opposed to single-agent optimization problems), this requires a completely novel approach, probably foregoing the hope of obtaining a closed-form global solution.

Instead, it would be more natural to focus on the invariant measure of the process near the problem's solution modes, similarly to our analysis in Section 3.4. This would allow the characterization of adversarial training methods near local saddle-points and would provide at least *some* theoretical insights on the behavior of GANs in realistic models. As shown in Fig. 7.1, experiments in simple GANs show that the invariant measure of the Fokker-Planck distribution has remarkable predictive power for the end-state of the trained network. If this can be proved rigorously, this would be a considerable contribution in our understanding of GANs.

BIBLIOGRAPHY

[1] Jacob Abernethy, Peter L. Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *COLT '08: Proceedings of the 21st Annual Conference on Learning Theory*, 2008.

[2] Alekh Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT '10: Proceedings of the 23rd Annual Conference on Learning Theory*, 2010.

[3] Ethan Akin. Domination or equilibrium. *Mathematical Biosciences*, 50(3-4):239–250, 1980.

[4] Felipe Alvarez, Jérôme Bolte, and Olivier Brahic. Hessian Riemannian gradient flows in convex programming. *SIAM Journal on Control and Optimization*, 43(2):477–501, 2004.

[5] Theodore Wilbur Anderson. *An Introduction to Multivariate Statistical analysis*. Wiley-Interscience, 3rd edition, 2003.

[6] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[7] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995.

[8] Robert J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, March 1974.

[9] Robert J. Aumann. Correlated equilibrium as an expression of Baeysian rationality. *Econometrica*, 55(1):1–18, 1987.

[10] David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *ICML '18: Proceedings of the 35th International Conference on Machine Learning*, 2018.

[11] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, NY, USA, 2 edition, 2017.

[12] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[13] E. Veronica Belmega, Panayotis Mertikopoulos, Romain Negrel, and Luca Sanguinetti. Online convex optimization and no-regret learning: Algorithms, guarantees and applications. `https://arxiv.org/abs/1804.04529`, 2018.

[14] Michel Benaïm. Dynamics of stochastic approximation algorithms. In Jacques Azéma, Michel Émery, Michel Ledoux, and Marc Yor, editors, *Séminaire de Probabilités XXXIII*, volume 1709 of *Lecture Notes in Mathematics*, pages 1–68. Springer Berlin Heidelberg, 1999.

[15] Michel Benaïm and Morris W. Hirsch. Asymptotic pseudotrajectories and chain recurrent flows, with applications. *Journal of Dynamics and Differential Equations*, 8(1):141–176, 1996.

[16] Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

[17] Dimitri P. Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1992.

[18] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 2015.

[19] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.

[20] Avrim Blum and Yishay Mansour. Learning, regret minimization, and equilibria. In Noam Nisan, Tim Roughgarden, Éva Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 4. Cambridge University Press, 2007.

[21] Jérôme Bolte and Marc Teboulle. Barrier operators and associated gradient-like dynamical systems for constrained minimization problems. *SIAM Journal on Control and Optimization*, 42(4):1266–1292, 2003.

[22] Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press and Hindustan Book Agency, 2008.

[23] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[24] Mario Bravo. An adjusted payoff-based procedure for normal form games. *Mathematics of Operations Research*, 41(4):1469–1483, November 2016.

[25] Mario Bravo and Panayotis Mertikopoulos. On the robustness of learning in games with stochastically perturbed payoff observations. *Games and Economic Behavior*, 103, John Nash Memorial issue:41–66, May 2017.

[26] Mario Bravo, David S. Leslie, and Panayotis Mertikopoulos. Bandit learning in concave N-person games. In *NIPS '18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[27] Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[28] William Brian Arthur. Inductive reasoning and bounded rationality (the El Farol problem). *American Economic Review*, 84(2):406–411, 1994.

[29] George W. Brown. Iterative solutions of games by fictitious play. In T. C. Coopmans, editor, *Activity Analysis of Productions and Allocation*, 374-376. Wiley, 1951.

[30] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[31] Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. In *COLT '16: Proceedings of the 29th Annual Conference on Learning Theory*, 2016.

[32] Sébastien Bubeck and Ronen Eldan. Kernel-based methods for bandit convex optimization. In *STOC '17: Proceedings of the 49th annual ACM SIGACT symposium on the Theory of Computing*, 2017.

[33] Antonio Cabrales. Stochastic replicator dynamics. *International Economic Review*, 41(2): 451–81, May 2000.

[34] Ozan Candogan, Ishai Menache, Asuman Ozdaglar, and Pablo A. Parrilo. Flows and decompositions of games: harmonic and potential games. *Mathematics of Operations Research*, 36(3):474–503, 2011.

[35] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[36] Nicolò Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, 31(3):562–580, April 2006.

[37] Gong Chen and Marc Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, August 1993.

[38] R. S. Cheng and Sergio Verdú. Gaussian multiaccess channels with ISI: capacity region and multiuser water-filling. *IEEE Trans. Inf. Theory*, 39(3):773–785, May 1993.

[39] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. Learning with bandit feedback in potential games. In *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[40] Johanne Cohen, Amélie Héliou, and Panayotis Mertikopoulos. Hedging under uncertainty: Regret minimization meets exponentially fast convergence. In *SAGT '17: Proceedings of the 10th International Symposium on Algorithmic Game Theory*, 2017.

[41] Roberto Cominetti, Emerson Melo, and Sylvain Sorin. A payoff-based learning procedure and its application to traffic games. *Games and Economic Behavior*, 70(1):71–83, 2010.

[42] Pierre Coucheney, Bruno Gaujal, and Panayotis Mertikopoulos. Penalty-regulated dynamics and robust learning procedures in games. *Mathematics of Operations Research*, 40(3):611–633, August 2015.

[43] Constantinos Daskalakis. *The complexity of Nash equilibria*. PhD thesis, University of California, Berkeley, 2008.

[44] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC '06: Proceedings of the 38th annual ACM symposium on the Theory of Computing*, 2006.

[45] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[46] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[47] A. M. Davie and Andrew J. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh, Section A: Mathematics*, 143(2):351–369, 4 2013.

[48] Gérard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences of the USA*, 38(10):886–893, October 1952.

[49] Gérard Debreu. Smooth preferences. *Econometrica*, 40(4):603–615, July 1972.

[50] Francisco Facchinei and Jong-Shi Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research. Springer, 2003.

[51] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA '05: Proceedings of the 16th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[52] Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59(3):637–666, 1991.

[53] Drew Fudenberg and Christopher Harris. Evolutionary dynamics with aggregate shocks. *Journal of Economic Theory*, 57(2):420–441, August 1992.

[54] Drew Fudenberg and David K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19(5-7):1065–1089, 1995.

[55] Itzhak Gilboa and Akihiko Matsui. Social stability and equilibrium. *Econometrica*, 59(3): 859–867, May 1991.

[56] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS '14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014.

[57] James Hannan. Approximation to Bayes risk in repeated play. In Melvin Dresher, Albert William Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Volume III*, volume 39 of *Annals of Mathematics Studies*, pages 97–139. Princeton University Press, Princeton, NJ, 1957.

[58] Marc Harper. Escort evolutionary game theory. *Physica D: Nonlinear Phenomena*, 240(18): 1411–1415, September 2011.

[59] John Charles Harsanyi. Oddness of the number of equilibrium points: a new proof. *International Journal of Game Theory*, 2(1):235–250, December 1973.

[60] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, September 2000.

[61] Sergiu Hart and Andreu Mas-Colell. A reinforcement procedure leading to correlated equilibrium. In *Economic Essays*, pages 181–200. Springer-Verlag, Berlin, 2001.

[62] Sergiu Hart and Andreu Mas-Colell. Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, 93(5):1830–1836, 2003.

[63] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, December 2007.

[64] Josef Hofbauer and Lorens A. Imhof. Time averages, recurrence and transience in the stochastic replicator dynamics. *The Annals of Applied Probability*, 19(4):1347–1368, 2009.

[65] Josef Hofbauer and William H. Sandholm. Stable games and their dynamics. *Journal of Economic Theory*, 144(4):1665–1693, July 2009.

[66] Josef Hofbauer and Karl Sigmund. Adaptive dynamics and evolutionary stability. *Applied Mathematics Letters*, 3:75–79, 1990.

[67] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK, 1998.

[68] Josef Hofbauer, Peter Schuster, and Karl Sigmund. A note on evolutionarily stable strategies and game dynamics. *Journal of Theoretical Biology*, 81(3):609–612, 1979.

[69] Josef Hofbauer, Sylvain Sorin, and Yannick Viossat. Time average replicator and best reply dynamics. *Mathematics of Operations Research*, 34(2):263–269, May 2009.

[70] Ed Hopkins. Learning, matching, and aggregation. *Games and Economic Behavior*, 26: 79–110, 1999.

[71] Lorens A. Imhof. The long-run behavior of the stochastic replicator dynamics. *The Annals of Applied Probability*, 15(1B):1019–1045, 2005.

[72] J. Jose, A. Ashikhmin, Thomas L. Marzetta, and Sriram Vishwanath. Pilot contamination and precoding in multi-cell TDD systems. *IEEE Trans. Wireless Commun.*, 10(8):2640–2651, 2011.

[73] Anatoli Juditsky, Arkadi Semen Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[74] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, Berlin, 1998.

[75] Frank P. Kelly, Aman K. Maulloo, and David K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, March 1998.

[76] Rafail Z. Khasminskii. *Stochastic Stability of Differential Equations*. Number 66 in Stochastic Modelling and Applied Probability. Springer-Verlag, Berlin, 2 edition, 2012.

[77] Krzysztof C. Kiwiel. Free-steering relaxation methods for problems with strictly convex costs and linear constraints. *Mathematics of Operations Research*, 22(2):326–349, 1997.

[78] Robert Kleinberg, Georgios Piliouras, and Éva Tardos. Load balancing without regret in the bulletin board model. *Distributed Computing*, 24(1):21–29, 2011.

[79] Robert D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *NIPS' 04: Proceedings of the 18th Annual Conference on Neural Information Processing Systems*, 2004.

[80] Joon Kwon and Panayotis Mertikopoulos. A continuous-time approach to online optimization. *Journal of Dynamics and Games*, 4(2):125–148, April 2017.

[81] Ratul Lahkar and William H. Sandholm. The projection dynamic and the geometry of population games. *Games and Economic Behavior*, 64:565–590, 2008.

[82] Rida Laraki and Panayotis Mertikopoulos. Higher order game dynamics. *Journal of Economic Theory*, 148(6):2666–2695, November 2013.

[83] Rida Laraki, Jérôme Renault, and Sylvain Sorin. *Mathematical Foundations of Game Theory*. Universitext. Springer, 2019.

[84] David S. Leslie and E. J. Collins. Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, 44(2):495–514, 2005.

[85] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[86] Lennart Ljung. Strong convergence of a stochastic approximation algorithm. *Annals of Statistics*, 6(3):680–696, 1978.

[87] Jason R. Marden and Jeff S. Shamma. Game theory and distributed control. In H. Peyton Young and Shmuel Zamir, editors, *Handbook of Game Theory*, volume 4, pages 861–899. Elsevier, 2015.

[88] John Maynard Smith and George R. Price. The logic of animal conflict. *Nature*, 246:15–18, November 1973.

[89] Ruta Mehta, Ioannis Panageas, and Georgios Piliouras. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics. In *ITCS '15: Proceedings of the 6th Conference on Innovations in Theoretical Computer Science*, 2015.

[90] Panayotis Mertikopoulos and Aris L. Moustakas. The emergence of rational behavior in the presence of stochastic perturbations. *The Annals of Applied Probability*, 20(4):1359–1388, July 2010.

[91] Panayotis Mertikopoulos and Aris L. Moustakas. Learning in an uncertain world: MIMO covariance matrix optimization with imperfect feedback. *IEEE Trans. Signal Process.*, 64(1): 5–18, January 2016.

[92] Panayotis Mertikopoulos and William H. Sandholm. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4):1297–1324, November 2016.

[93] Panayotis Mertikopoulos and William H. Sandholm. Riemannian game dynamics. *Journal of Economic Theory*, 177:315–364, September 2018.

[94] Panayotis Mertikopoulos and Mathias Staudigl. Convergence to Nash equilibrium in continuous games with noisy first-order feedback. In *CDC '17: Proceedings of the 56th IEEE Annual Conference on Decision and Control*, 2017.

[95] Panayotis Mertikopoulos and Mathias Staudigl. On the convergence of gradient-like flows with noisy gradient input. *SIAM Journal on Optimization*, 28(1):163–197, January 2018.

[96] Panayotis Mertikopoulos and Yannick Viossat. Imitation dynamics with payoff shocks. *International Journal of Game Theory*, 45(1-2):291–320, March 2016.

[97] Panayotis Mertikopoulos and Zhengyuan Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, January 2019.

[98] Panayotis Mertikopoulos, E. Veronica Belmega, and Aris L. Moustakas. Matrix exponential learning: Distributed optimization in MIMO systems. In *ISIT '12: Proceedings of the 2012 IEEE International Symposium on Information Theory*, pages 3028–3032, 2012.

[99] Panayotis Mertikopoulos, E. Veronica Belmega, Aris L. Moustakas, and Samson Lasaulce. Distributed learning policies for power allocation in multiple access channels. *IEEE J. Sel. Areas Commun.*, 30(1):96–106, January 2012.

[100] Panayotis Mertikopoulos, E. Veronica Belmega, Romain Negrel, and Luca Sanguinetti. Distributed stochastic optimization via matrix exponential learning. *IEEE Trans. Signal Process.*, 65(9):2277–2290, May 2017.

[101] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *SODA '18: Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[102] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *ICLR '19: Proceedings of the 2019 International Conference on Learning Representations*, 2019.

[103] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14 (1):124 – 143, 1996.

[104] Hervé Moulin and Jean-Philippe Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7:201–221, 1978.

[105] John H. Nachbar. Evolutionary selection dynamics in games. *International Journal of Game Theory*, 19:59–89, 1990.

[106] A. Nagurney and D. Zhang. Projected dynamical systems in the formulation, stability analysis, and computation of fixed demand traffic network equilibria. *Transportation Science*, 31:147–158, 1997.

[107] John F. Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, September 1951.

[108] Arkadi Semen Nemirovski and David Berkovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, NY, 1983.

[109] Arkadi Semen Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[110] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2):319–344, 2007.

[111] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

[112] Hukukane Nikaido and Kazuo Isoda. Note on non-cooperative convex games. *Pacific Journal of Mathematics*, 5:807–815, 1955.

[113] Noam Nisan, Tim Roughgarden, Éva Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[114] A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

[115] Ariel Orda, Raphael Rom, and Nahum Shimkin. Competitive routing in multi-user communication networks. *IEEE/ACM Trans. Netw.*, 1(5):614–627, October 1993.

[116] Boris Teodorovich Polyak. *Introduction to Optimization*. Optimization Software, New York, NY, USA, 1987.

[117] Boris Teodorovich Polyak and Anatoli Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, July 1992.

[118] Maxim Raginsky and Jake Bouvrie. Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence. In *CDC '13: Proceedings of the 52nd IEEE Annual Conference on Decision and Control*, 2013.

[119] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[120] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[121] Julia Robinson. An iterative method for solving a game. *Annals of Mathematics*, 54:296–301, 1951.

[122] J. B. Rosen. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica*, 33(3):520–534, 1965.

[123] Howard Harry Rosenbrock. An automatic method for finding the greatest or least value of a function. *Computer Journal*, 3(3):175–184, 1960.

[124] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[125] Tim Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, Cambridge, MA, USA, 2005.

[126] Aldo Rustichini. Optimal properties of stimulus-response learning models. *Games and Economic Behavior*, 29(1-2):244–273, 1999.

[127] Larry Samuelson and Jianbo Zhang. Evolutionary stability in asymmetric games. *Journal of Economic Theory*, 57:363–391, 1992.

[128] William H. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97:81–108, 2001.

[129] William H. Sandholm. *Population Games and Evolutionary Dynamics*. MIT Press, Cambridge, MA, 2010.

[130] William H. Sandholm. Population games and deterministic evolutionary dynamics. In H. Peyton Young and Shmuel Zamir, editors, *Handbook of Game Theory IV*, pages 703–778. Elsevier, 2015.

[131] William H. Sandholm, Emin Dokumacı, and Ratul Lahkar. The projection dynamic and the replicator dynamic. *Games and Economic Behavior*, 64:666–683, 2008.

[132] Gesualdo Scutari, Daniel Pérez Palomar, and Sergio Barbarossa. Simultaneous iterative water-filling for Gaussian frequency-selective interference channels. In *ISIT '06: Proceedings of the 2006 International Symposium on Information Theory*, 2006.

[133] Gesualdo Scutari, Daniel Pérez Palomar, and Sergio Barbarossa. The MIMO iterative waterfilling algorithm. *IEEE Trans. Signal Process.*, 57(5):1917–1935, May 2009.

[134] Gesualdo Scutari, Francisco Facchinei, Daniel Pérez Palomar, and Jong-Shi Pang. Convex optimization, game theory, and variational inequality theory in multiuser communication systems. *IEEE Signal Process. Mag.*, 27(3):35–49, May 2010.

[135] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University of Jerusalem, 2007.

[136] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[137] Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and Fenchel duality. In *Advances in Neural Information Processing Systems 19*, pages 1265–1272. MIT Press, 2007.

[138] Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *COLT '13: Proceedings of the 26th Annual Conference on Learning Theory*, 2013.

[139] Sylvain Sorin. Exponential weight algorithm in continuous time. *Mathematical Programming*, 116(1):513–528, 2009.

[140] Sylvain Sorin and Cheng Wan. Finite composite games: Equilibria and dynamics. *Journal of Dynamics and Games*, 3(1):101–120, January 2016.

[141] James C. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1):109–112, 1997.

[142] Peter D. Taylor. Evolutionarily stable strategies with two types of player. *Journal of Applied Probability*, 16(1):76–83, March 1979.

[143] Peter D. Taylor and Leo B. Jonker. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1-2):145–156, 1978.

[144] I. Emre Telatar. Capacity of multi-antenna Gaussian channels. *European Transactions on Telecommunications and Related Technologies*, 10(6):585–596, 1999.

[145] John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Autom. Control*, 31(9):803–812, 1986.

[146] Gordon Tullock. Efficient rent seeking. In J. M. Buchanan R. D. Tollison and Gordon Tullock, editors, *Toward a theory of the rent-seeking society*. Texas A&M University Press, 1980.

[147] Luigi Vigneri, Georgios Paschos, and Panayotis Mertikopoulos. Large-scale network utility maximization: Countering exponential growth with exponentiated gradients. In *INFOCOM '19: Proceedings of the 38th IEEE International Conference on Computer Communications*, 2019.

[148] Yannick Viossat and Andriy Zapechelnyuk. No-regret dynamics and fictitious play. *Journal of Economic Theory*, 148(2):825–842, March 2013.

[149] John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100: 295–320, 1928. Translated by S. Bargmann as "On the Theory of Games of Strategy" in A. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV*, volume 40 of *Annals of Mathematics Studies*, pages 13-42, 1957, Princeton University Press, Princeton.

[150] Vladimir G. Vovk. Aggregating strategies. In *COLT '90: Proceedings of the 3rd Workshop on Computational Learning Theory*, pages 371–383, 1990.

[151] Jörgen W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, MA, 1995.

[152] Wei Yu, Wonjong Rhee, Stephen Boyd, and John M. Cioffi. Iterative water-filling for Gaussian vector multiple-access channels. *IEEE Trans. Inf. Theory*, 50(1):145–152, 2004.

[153] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W. Glynn. Stochastic mirror descent for variationally coherent optimization problems. In *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[154] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Peter W. Glynn, and Yinyu Ye. Distributed stochastic optimization with large delays. Under review, 2018.

[155] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Peter W. Glynn, Yinyu Ye, Jia Li, and Fei-Fei Li. Distributed asynchronous optimization with unbounded delays: How slow can you go? In *ICML '18: Proceedings of the 35th International Conference on Machine Learning*, 2018.

[156] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W. Glynn. On the convergence of mirror descent beyond stochastic convex programming. *SIAM Journal on Optimization*, forthcoming.

[157] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML '03: Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

APPENDIX

A

VITÆ

EDUCATION AND PROFESSIONAL EXPERIENCE

2011–present	**CNRS – Laboratoire d'Informatique de Grenoble** *Chargé de recherche, CN* (senior researcher) at LIG	Grenoble, FR
2019, fall	**École Polytechnique Fédérale de Lausanne (EPFL)** Visiting professor for the fall semester of 2019-2020	Lausanne, CH
2018, spring	**UC Berkeley** Visiting Scientist for the spring semester of 2017-2018	Berkeley, CA, USA
2016, fall	**LUISS Guido Carli University** Visiting professor for the fall semester of 2016-2017	Rome, IT
2010–2011	**École Polytechnique** Post-doctoral researcher in game theory	Paris, FR
2007–2010	**University of Athens, Department of Physics** *Doctorate of Philosophy* "*Stochastic perturbations in game theory and applications to networks*"	Athens, GR
2003–2006	**Brown University** *Master of Science* in Mathematics (May 2005), *summa cum laude* *M. Phil.* in Mathematics (Sept. 2005)	Providence, RI, USA
1998–2003	**University of Athens** *Ptychion* degree in Physics (July 2003); valedictorian (9.1/10 GPA)	Athens, GR

AWARDS AND DISTINCTIONS

2018	Outstanding reviewer award at NeurIPS 2018
2017	INFORMS George Nicholson best student paper award finalist for "*Multi-agent online learning with imperfect information*"
2012	Best paper award at NETGCOOP 2012 for "*Strange bedfellows: Riemann, Gibbs, and vector Gaussian multiple access channels*"

GRANTS AND COLLABORATIONS

Awarded grants

2016–2020 **ANR ORACLESS** **Budget: 207 k€**

Online resource allocation for unpredictable large-scale wireless systems

ANR starting grant (JCJC), PI.

This project is an ANR starting grant aiming to develop adaptive resource allocation schemes that are provably capable of tracking unpredictable changes in communication networks. As the project's PI, I am coordinating its research activities and scientific output. O. Bilenne was recruited as a post-doctoral fellow to work on learning for MIMO networks in September 2018 (co-supervised with E. V. Belmega).

2017–2020 **GAMENET** **Budget: 625 k€**

European Network for Game Theory

EU COST action; working group leader.

The European Network for Game Theory is an EU COST action initiated by M. Staudigl (Maastricht University, the Netherlands), M. Scarsini (LUISS, Rome), and myself. It is bringing together more than 20 countries, with the aim of promoting research in game theory and beyond. I am a founding member of GAMENET's core group (the main decision body within the the action's management committee), MC representative of France, and scientific coordinator of the activities of WG2 (working group on learning in distributed large-scale networks).

2017–2018 **ULTRON** **Budget: 195 k€**

Ultra-low latency scheduling via online learning

Huawei FLAGSHIP grant, PI.

The ULTRON project is a Huawei HIRP FLAGSHIP project aiming to develop highly adaptive learning policies for achieving ultra-low latencies in 5G mobile systems. Owing to the applications of my work on multi-agent learning to computer networks and communications, I was invited to submit this proposal in 2016. The project ran successfully for an 18-month life-cycle between 2017 and 2018 and led to novel resource allocation schemes for software-defined networks that are currently being utilized by Huawei. L. Vigneri and A. Héliou were recruited as post-doctoral fellows working on different aspects of this project.

2018 **MixedGAN** **Budget: 10 k€**

Mixed-strategy generative adversarial networks

CNRS exploratory grant (PEPS I3A); co-PI structure.

This PEPS project is coordinated by R. Laraki (LAMSADE) and has a co-PI structure.

2014–2017 **ANR GAGA** **Budget: 85 k€**

Geometric aspects of games

ANR grant; co-PI structure.

This project was an ANR starting grant aiming to explore the role of geometric structures in game theory and learning. It was coordinated by V. Perchet with a co-PI structure (typical of ANR grants in mathematics). I was responsible for coordinating the activities on Work-Package 2: "Geometric algorithms and dynamics for learning in games".

2017–2018 **HEAVY.NET** **Budget: 12 k€**
Optimization and analysis of heavily congested networks
PGMO/PRMO grant; PI

2016 **REAL.net** **Budget: 10 k€**
Resource allocation in dynamic network environments
CNRS exploratory grant (PEPS JCJC); PI

2014–2015 **GATHERING** **Budget: 10 k€**
Game theory, evolution and randomness in networks and graphs
CNRS exploratory grant (PEPS HuMaIn); PI

Participation in research projects and networks

2016–2018 **LEARN**
Learning algorithms for games and applications
Franco-Chilean Network of Excellence, co-financed by ECOS-Sud and CONICYT

2013–2017 **NETLEARN**
Learning algorithms orchestration for mobile networks resource management
Research project financed by the French National Research Agency (ANR)

2012–2015 **NEWCOM#**
Network of excellence in wireless communications
Network of Excellence formed under FP7

2012–2016 **ADGO**
Algorithms and dynamics in games and optimization
Franco-Chilean network funded by the Chilean National Research Agency (CONICYT)

2012–2016 **CROWN**
Optimal control of self-organized wireless networks
Research project co-financed by EU and Greek national funds under the THALES initiative

2006–2009 **NET-REFOUND**
Network research foundations and trends
Specific Targeted Research Project funded by the EU under FP6

Scientific stays abroad

2018 **UC Berkeley** Berkeley, CA, USA
I spent most of the 2017-2018 spring semester at the Simons Center for the Theory of Computing at UC Berkeley. I was invited there to participate in the "Real-Time Decision-Making" thematic program as an expert on multi-agent and game-theoretic learning.

2017 **Stanford Unviversity** Palo Alto, CA, USA
I spent a month in the 2016-2017 spring semester at the Computer Network Architecture and Performance Engineering Lab at Stanford University. This was part of an ongoing collaboration with N. Bambos and was an essential part of my mentoring (unofficial PhD supervision) of Z. Zhou.

2016 **LUISS Guido Carli University** Rome, Italy
I spent a month in the 2016-2017 fall semester at the Operations Research department of the LUISS Guido Carli University. This was part of an ongoing collaboration project with M. Scarsini and the starting point of the GAMENET proposal.

2014-2017 **University of Athens** Athens, Greece
Since 2014, I have been teaching several modules at the University of Athens (mostly on probability and advanced optimization algorithms for advanced undergraduate students), and I am actively collaborating with the Wireless Systems Lab of the University of Athens.

SCIENTIFIC AND ADMINISTRATIVE RESPONSIBILITIES

Coordination activities

2017–present Coordinator of the working group on learning (WG2) of the European Network for Game Theory (EU COST action GAMENET). I am also part of the action's core group, coordinating the overall activities of the network, and interfacing with COST at Brussels.

2014–present Member of the steering committee (*comité de liaison*) of the optimization and decision theory group of the French Society for Industrial and Applied Mathematics (SMAI). This group coordinates the activities of the SMAI in the areas of decision sciences, optimization, and game theory.

2011–present Graduate students liaison (*chargé de mission doctorants*) for the Laboratoire d'Informatique de Grenoble. I am acting as the official ombudsperson for graduate students, and I am organizing every year the PhD welcome day for new graduate students, and LIG's PhD day for second-year PhD students.

Editorial activities

Boards Associate editor for JDG (Journal on Dynamics and Games) and MCAP (Methodology and Computing in Applied Probability)

Chairing Area chair for NeurIPS (2019)

Reviewing (journals) *Advances in Applied Probability*, *Annals of Operations Research*, *Dynamic Games and Applications*, *Games and Economic Behavior*, *IEEE Access*, *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Information Theory / Signal Processing / Communications / Wireless Communications*, *IEEE/ACM Transactions on Networking*, *Journal of Economic Theory*, *Journal of Optimization Theory and Applications*, *Mathematics of Operations Research*, *Mathematical Programming*, *Operations Research*, *SIAM Journal on Control and Optimization*, *SIAM Journal on Optimization*, *Theoretical Economics*, …

Reviewing (conferences)	COLT, ICML, NeurIPS, SODA, …

Conference organization

2019	Co-organizer of the workshop "Twenty years of the Price of Anarchy"
2018	Co-organizer of the 2018 Paris Symposium on Game Theory
2018	General co-chair of the French Days on Optimization and Decision Science ("*Journées SMAI–MODE 2018*")
2018	Co-organizer of the 2018 Workshop on "Games, Dynamics and Optimization"
2016	General co-chair of the 2016 Workshop on "*Geometry, Evolution and Learning in Games*")
2015	Organizer of the mini-symposium "*Games, Learning and Applications*" at the 2015 SMAI congress
2014	Program co-chair of the 12th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks
2013	General co-chair of the 2013 Intl. Workshop on Algorithmic Game Theory
2013	Publications chair of the 11th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2013)

Committees

2018	Adil Salim, TELECOM ParisTech, external examiner. "*Random monotone operators and application to stochastic optimization*"
2014	Tatiana Seregina, Université de Toulouse, external examiner. "*Applications of game theory to distributed routing and delay-tolerant networking*"

RESEARCH SUPERVISION AND TEACHING

Post-docs

- Olivier Bilenne (2018–present)
  Topic: online optimization for MIMO [ANR ORACLESS]
- Amélie Héliou (2017–2018)
  Topic: scalable routing algorithms [ULTRON project]
- Luigi Vigneri (2017–2018)
  Topic: scalable latency minimization [ULTRON project]
- Ioannis Stiakogiannakis (2014–2015)
  Topic: game theory for MIMO systems [ANR NETLEARN]
- Nof Abuzainab (2014–2015)
  Topic: dynamic spectrum access [funded by Inria]

Ph.D. students

- Benjamin Roussillon
  2018—, UGA; co-supervised w. P. Loiseau
  Topic: "*Classification en présence de données adverses : modèles et solutions*"
- Kimon Antonakopoulos
  2017—, UGA; co-supervised with E. V. Belmega
  Topic: "Online learning in variational inequality problems"
- Bruno Donassolo
  2017—, Orange; co-supervised with A. Legrand and I. Fajjari
  Topic: "*Decentralized management of applications in Fog computing environments*"
- Alexandre Marcastel
  2015–2019, ENSEA; co-supervised with E. V. Belmega and I. Fijalkow
  Topic: "*Allocation de puissance en ligne dans un réseau IoT dynamique et non-prédictible*"
- Zhengyuan Zhou
  2014–2019, Stanford University; mentored under the supervision of N. Bambos and P. W. Glynn
  Topic: "*Multi-agent online decision-making with imperfect feedback: Theory and applications*"

Teaching

- ENS Lyon: game theory, learning, optimization (*cours magistral*)
- Trinity College Dublin (2019): Summer school on online optimization for wireless systems
- UC Berkeley (2018): real-time decision-making
- RESCOM (2012): Summer school on the applications of game theory to data networks

INVITED TALKS AND TUTORIALS (PAST 3 YEARS ONLY)

I am regularly invited to give talks in conferences, workshops and top-tier research institutions / universities. I am providing below a non-exhaustive list of invited talks and tutorials in the past 3 years:

2019 **CONNECT Summer School on Machine Learning** Dublin, Ireland
"*Online learning and optimization for wireless systems*"

2019 **NPCG 2019 – Networks and Congestion Games** Paris, France
"*No-regret learning in games*"

2019 **GDO 2019 – Games, Dynamics and Optimization** Cluj-Napoca, Romania
"*Hessian barrier algorithms for linearly constrained optimization problems*"

2019 **OSL 2019 – Optimization and Statistical Learning** Les Houches, France
"*Extra-gradient methods for variational inequalities*"

2019 **EPFL** Lausanne, Switzerland
"*Going the extra (gradient) mile in GAN training*"

2019 **Criteo AI Lab** Paris, France
"*Applications of multi-agent learning to computational advertising*"

2018 **PGMO Days 2018** Paris, France
"*Learning dynamics for routing problems*"

2018 **Trinity College Dublin** Dublin, Ireland
"*Efficient network utility maximization algorithms*"

2018 **National Technical University of Athens** Athens, Greece
"*Traffic in congested networks: Equilibrium, efficiency, and dynamics*"

2018 **GDO 2018** Vienna, Austria
"*Bandit learning in concave N-person games*"

2018 **Google Inc.** Mountain View, CA, USA
"*Accelerated and optimistic methods for learning*"

2018 **UC Berkeley – Simons Institute** Berkeley, CA, USA
"*Online learning in games*"

2017 **University of Aix–Marseille** Marseille, France
"*Convergence and non-convergence in game-theoretic learning*"

2017 **PGMO Days 2017** Paris, France
"*The price of anarchy in high and low traffic*"

2017 **GDR ISIS workshop on Game Theory and Learning** Paris, France
"*Game theory meets signal processing (and feels no regret)*"

2017 **Emergent and Self-Adaptive Systems Workshop panel** Lancaster, UK
"*Design and validation of future computer systems: Theory and practice*"

2017 **Lancaster University** Lancaster, UK
"*Multi-agent online learning: Game theory meets machine learning*"

2017 **Paris Game Theory Seminar** Paris, France
"*No-regret learning in games*"

2017 **Erice 2017 – Stochastic Methods in Game Theory** Erice, Italy
"*How bad is selfish routing in highly congested networks?*"

2017 **Stanford University** Stanford, CA, USA
"*Learning in games via reinforcement and regularization*"

2016 **University of Vienna** Vienna, Austria
"*On the convergence of gradient flows with noisy gradient input*"

2016 **Saclay Algorithmics Seminar (Univ. Paris–Sud)** Paris, France
"*Learning in games with continuous action spaces*"

2016 **Sapienza University of Rome** Rome, Italy
"*Game-theoretic learning with noisy first-order input*"

2016 **LUISS Guido Carli University** Rome, Italy
"*Learning in games with imperfect information*"

2016 **Paris Optimization Seminar** Paris, France
"*Learning in concave games*"

2016 **CROWNCOM 2016 (two-part tutorial)** Grenoble, France
"*Game theory, learning, and cognitive radio*"

2016 **ADGO 2016 – Algorithms and Dynamics for Games and Optimization** Santiago, Chile
"*Robust optimization and online learning in games*"

B

PUBLICATIONS AND SCIENTIFIC OUTPUT

This appendix provides an up-to-date list of the author's scientific output and co-authored publications. Author ordering depends on the usual conventions of each field and with whom the article was written. It is usually alphabetical, unless there is a junior researcher who played a major part in the work (in which case their name often appears first).

WORKING / SUBMITTED PAPERS (7)

[1] R. I. Boţ, P. Mertikopoulos, M. Staudigl, and P. T. Vuong, "Forward-backward-forward methods with variance reduction for stochastic variational inequalities." `https://arxiv.org/abs/1902.03355`, 2019.

[2] Z. Zhou, P. Mertikopoulos, N. Bambos, P. W. Glynn, and Y. Ye, "Distributed stochastic optimization with large delays." Under review, 2018.

[3] B. Duvocelle, P. Mertikopoulos, M. Staudigl, and D. Vermeulen, "Learning in time-varying games." `https://arxiv.org/abs/1809.03066`, 2018.

[4] Z. Zhou, P. Mertikopoulos, N. Bambos, P. W. Glynn, and C. Tomlin, "Multi-agent online learning with imperfect information." Under review, 2018.

[5] E. V. Belmega, P. Mertikopoulos, R. Negrel, and L. Sanguinetti, "Online convex optimization and no-regret learning: Algorithms, guarantees and applications." `https://arxiv.org/abs/1804.04529`, 2018.

[6] I. Stiakogiannakis, P. Mertikopoulos, and C. Touati, "Power control via online learning in non-stationary MIMO networks." `http://arxiv.org/abs/1503.02155`, 2018.

[7] P. Mertikopoulos, A. L. Moustakas, and A. Tzanakaki, "Boltzmann meets Nash: Energy-efficient routing in optical networks under uncertainty." `https://arxiv.org/abs/1605.01451`, 2016.

JOURNAL PAPERS (32)

[1] Z. Zhou, P. Mertikopoulos, N. Bambos, S. Boyd, and P. W. Glynn, "On the convergence of mirror descent beyond stochastic convex programming," *SIAM Journal on Optimization*, forthcoming.

[2] R. Colini-Baldeschi, R. Cominetti, P. Mertikopoulos, and M. Scarsini, "When is selfish routing bad? The price of anarchy in light and heavy traffic," *Operations Research*, forthcoming.

[3] Z. Zhou, P. Mertikopoulos, A. L. Moustakas, N. Bambos, and P. W. Glynn, "Robust power management via learning and game design," *Operations Research*, forthcoming.

[4] I. M. Bomze, P. Mertikopoulos, W. Schachinger, and M. Staudigl, "Hessian barrier algorithms for linearly constrained optimization problems," *SIAM Journal on Optimization*, vol. 29, pp. 2100–2127, August 2019.

[5] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online power optimization in feedback-limited, dynamic and unpredictable IoT networks," *IEEE Trans. Signal Process.*, vol. 67, pp. 2987–3000, June 2019.

[6] P. Mertikopoulos and Z. Zhou, "Learning in games with continuous action sets and unknown payoff functions," *Mathematical Programming*, vol. 173, pp. 465–507, January 2019.

[7] P. Mertikopoulos and M. Staudigl, "Stochastic mirror descent dynamics and their convergence in monotone variational inequalities," *Journal of Optimization Theory and Applications*, vol. 179, pp. 838–867, December 2018.

[8] P. Mertikopoulos and W. H. Sandholm, "Riemannian game dynamics," *Journal of Economic Theory*, vol. 177, pp. 315–364, September 2018.

[9] P. Mertikopoulos and M. Staudigl, "On the convergence of gradient-like flows with noisy gradient input," *SIAM Journal on Optimization*, vol. 28, pp. 163–197, January 2018.

[10] P. Mertikopoulos, E. V. Belmega, R. Negrel, and L. Sanguinetti, "Distributed stochastic optimization via matrix exponential learning," *IEEE Trans. Signal Process.*, vol. 65, pp. 2277–2290, May 2017.

[11] M. Bravo and P. Mertikopoulos, "On the robustness of learning in games with stochastically perturbed payoff observations," *Games and Economic Behavior*, vol. 103, John Nash Memorial issue, pp. 41–66, May 2017.

[12] J. Kwon and P. Mertikopoulos, "A continuous-time approach to online optimization," *Journal of Dynamics and Games*, vol. 4, pp. 125–148, April 2017.

[13] S. D'Oro, L. Galluccio, P. Mertikopoulos, G. Morabito, and S. Palazzo, "Auction-based resource allocation in OpenFlow multi-tenant networks," *Computer Networks*, vol. 115, pp. 29–41, March 2017.

[14] A. S. Shafigh, P. Mertikopoulos, S. Glisic, and Y. M. Fang, "Semi-cognitive radio networks: A novel dynamic spectrum sharing mechanism," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 3, pp. 97–111, March 2017.

[15] S. Perkins, P. Mertikopoulos, and D. S. Leslie, "Mixed-strategy learning with continuous action sets," *IEEE Trans. Autom. Control*, vol. 62, pp. 379–384, January 2017.

[16] P. Mertikopoulos and W. H. Sandholm, "Learning in games via reinforcement and regularization," *Mathematics of Operations Research*, vol. 41, pp. 1297–1324, November 2016.

[17] A. L. Moustakas, P. Mertikopoulos, and N. Bambos, "Power optimization in random wireless networks," *IEEE Trans. Inf. Theory*, vol. 62, pp. 5030–5058, September 2016.

[18] B. Gaujal and P. Mertikopoulos, "A stochastic approximation algorithm for stochastic semidefinite programming," *Probability in the Engineering and Informational Sciences*, vol. 30, pp. 431–454, July 2016.

[19] P. Mertikopoulos and E. V. Belmega, "Learning to be green: Robust energy efficiency maximization in dynamic MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 34, pp. 743 – 757, April 2016.

[20] P. Mertikopoulos and Y. Viossat, "Imitation dynamics with payoff shocks," *International Journal of Game Theory*, vol. 45, pp. 291–320, March 2016.

[21] P. Mertikopoulos and A. L. Moustakas, "Learning in an uncertain world: MIMO covariance matrix optimization with imperfect feedback," *IEEE Trans. Signal Process.*, vol. 64, pp. 5–18, January 2016.

[22] S. D'Oro, P. Mertikopoulos, A. L. Moustakas, and S. Palazzo, "Interference-based pricing for opportunistic multi-carrier cognitive radio systems," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 6536–6549, December 2015.

[23] R. Laraki and P. Mertikopoulos, "Inertial game dynamics and applications to constrained optimization," *SIAM Journal on Control and Optimization*, vol. 53, pp. 3141–3170, October 2015.

[24] G. Bacci, E. V. Belmega, P. Mertikopoulos, and L. Sanguinetti, "Energy-aware competitive power allocation for heterogeneous networks under QoS constraints," *IEEE Trans. Wireless Commun.*, vol. 14, pp. 4728–4742, September 2015.

[25] P. Coucheney, B. Gaujal, and P. Mertikopoulos, "Penalty-regulated dynamics and robust learning procedures in games," *Mathematics of Operations Research*, vol. 40, pp. 611–633, August 2015.

[26] P. Mertikopoulos and E. V. Belmega, "Transmit without regrets: online optimization in MIMO–OFDM cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 32, pp. 1987–1999, November 2014.

[27] R. Laraki and P. Mertikopoulos, "Higher order game dynamics," *Journal of Economic Theory*, vol. 148, pp. 2666–2695, November 2013.

[28] P. Mertikopoulos, E. V. Belmega, A. L. Moustakas, and S. Lasaulce, "Distributed learning policies for power allocation in multiple access channels," *IEEE J. Sel. Areas Commun.*, vol. 30, pp. 96–106, January 2012.

[29] C. Pawlowitsch, P. Mertikopoulos, and N. Ritt, "Neutral stability, drift, and the diversification of languages," *Journal of Theoretical Biology*, vol. 287, pp. 1–12, July 2011.

[30] P. Kazakopoulos, P. Mertikopoulos, A. L. Moustakas, and G. Caire, "Living at the edge: a large deviations approach to the outage MIMO capacity," *IEEE Trans. Inf. Theory*, vol. 57, pp. 1984–2007, April 2011.

[31] P. Mertikopoulos and A. L. Moustakas, "The emergence of rational behavior in the presence of stochastic perturbations," *The Annals of Applied Probability*, vol. 20, pp. 1359–1388, July 2010.

[32] P. Mertikopoulos and A. L. Moustakas, "Correlated anarchy in overlapping wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 26, pp. 1160–1169, September 2008.

CONFERENCE PROCEEDINGS (51)

[1] K. Antonakopoulos, E. V. Belmega, and P. Mertikopoulos, "Online and stochastic optimization beyond Lipschitz continuity: A Riemannian approach," in *ICLR '20: Proceedings of the 2020 International Conference on Learning Representations*, 2020.

[2] K. Antonakopoulos, E. V. Belmega, and P. Mertikopoulos, "An adaptive mirror-prox algorithm for variational inequalities with singular operators," in *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[3] Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos, "On the convergence of single-call stochastic extra-gradient methods," in *NeurIPS '19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[4] B. Donassolo, I. Fajjari, A. Legrand, and P. Mertikopoulos, "A fog-based framework for IoT service provisioning," in *CCNC '19: Proceedings of the 16th IEEE International Conference on Consumer Communications & Networking*, 2019.

[5] N. Liakopoulos, A. Destounis, G. Paschos, T. Spyropoulos, and P. Mertikopoulos, "Cautious regret minimization: Online optimization with long-term budget constraints," in *ICML '19: Proceedings of the 36th International Conference on Machine Learning*, 2019.

[6] M. Staudigl and P. Mertikopoulos, "Convergent noisy forward-backward-forward algorithms in non-monotone variational inequalities," in *LSS '19: Proceedings of the 15th IFAC Symposium on Large Scale Complex Systems*, 2019.

[7] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Gradient-free online resource allocation algorithms for dynamic wireless networks," in *SPAWC '19: Proceedings of the 2019 IEEE International Workshop on Signal Processing Advances in Wireless Communications*, 2019.

[8] L. Vigneri, G. Paschos, and P. Mertikopoulos, "Large-scale network utility maximization: Countering exponential growth with exponentiated gradients," in *INFOCOM '19: Proceedings of the 38th IEEE International Conference on Computer Communications*, 2019.

[9] I. Fajjari, B. Donassolo, A. Legrand, and P. Mertikopoulos, "Load-aware provisioning of IoT services on fog computing platform," in *ICC '19: Proceedings of the 2019 IEEE International Conference on Communications*, 2019.

[10] P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras, "Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile," in *ICLR '19: Proceedings of the 2019 International Conference on Learning Representations*, 2019.

[11] M. Leconte, G. Paschos, P. Mertikopoulos, and U. Kozat, "A resource allocation framework for network slicing," in *INFOCOM '18: Proceedings of the 37th IEEE International Conference on Computer Communications*, 2018.

[12] M. Bravo, D. S. Leslie, and P. Mertikopoulos, "Bandit learning in concave N-person games," in *NIPS '18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[13] P. Mertikopoulos, C. H. Papadimitriou, and G. Piliouras, "Cycles in adversarial regularized learning," in *SODA '18: Proceedings of the 29th annual ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[14] Z. Zhou, P. Mertikopoulos, N. Bambos, P. W. Glynn, Y. Ye, J. Li, and F.-F. Li, "Distributed asynchronous optimization with unbounded delays: How slow can you go?," in *ICML '18: Proceedings of the 35th International Conference on Machine Learning*, 2018.

[15] Z. Zhou, P. Mertikopoulos, S. Athey, N. Bambos, P. W. Glynn, and Y. Ye, "Learning in games with lossy feedback," in *NIPS '18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[16] R. I. Boţ, P. Mertikopoulos, M. Staudigl, and P. T. Vuong, "On the convergence of stochastic forward-backward-forward algorithms with variance reduction in pseudo-monotone variational inequalities," in *NIPS' 18: Workshop on Smooth Games, Optimization and Machine Learning (SGO&ML)*, 2018.

[17] A. Ward, Z. Zhou, P. Mertikopoulos, and N. Bambos, "Power control with random delays: Robust feedback averaging," in *CDC '18: Proceedings of the 57th IEEE Annual Conference on Decision and Control*, 2018.

[18] P. Mertikopoulos and M. Staudigl, "Convergence to Nash equilibrium in continuous games with noisy first-order feedback," in *CDC '17: Proceedings of the 56th IEEE Annual Conference on Decision and Control*, 2017.

[19] Z. Zhou, P. Mertikopoulos, N. Bambos, P. W. Glynn, and C. Tomlin, "Countering feedback delays in multi-agent learning," in *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[20] J. Cohen, A. Héliou, and P. Mertikopoulos, "Hedging under uncertainty: Regret minimization meets exponentially fast convergence," in *SAGT '17: Proceedings of the 10th International Symposium on Algorithmic Game Theory*, 2017.

[21] J. Cohen, A. Héliou, and P. Mertikopoulos, "Learning with bandit feedback in potential games," in *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[22] A. L. Moustakas, P. Mertikopoulos, Z. Zhou, and N. Bambos, "Least action routing: Identifying the optimal path in a wireless relay network," in *PIMRC'17: 28th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2017.

[23] Z. Zhou, P. Mertikopoulos, A. L. Moustakas, N. Bambos, and P. W. Glynn, "Mirror descent learning in continuous games," in *CDC '17: Proceedings of the 56th IEEE Annual Conference on Decision and Control*, 2017.

[24] Z. Zhou, P. Mertikopoulos, A. L. Moustakas, S. Mehdian, N. Bambos, and P. W. Glynn, "Power control in wireless networks via dual averaging," in *GLOBECOM '17: Proceedings of the 2017 IEEE Global Telecommunications Conference*, 2017.

[25] Z. Zhou, P. Mertikopoulos, N. Bambos, S. Boyd, and P. W. Glynn, "Stochastic mirror descent for variationally coherent optimization problems," in *NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[26] R. Colini-Baldeschi, R. Cominetti, P. Mertikopoulos, and M. Scarsini, "The asymptotic behavior of the price of anarchy," in *WINE 2017: Proceedings of the 13th Conference on Web and Internet Economics*, 2017.

[27] A. S. Shafigh, P. Mertikopoulos, and S. Glisic, "A novel dynamic network architecture model based on stochastic geometry and game theory," in *ICC '16: Proceedings of the 2016 IEEE International Conference on Communications*, 2016.

[28] P. Mertikopoulos, E. V. Belmega, and L. Sanguinetti, "Distributed learning for resource allocation under uncertainty," in *GLOBALSIP '16: Proceedings of the 2016 IEEE Global Conference on Signal and Information Processing*, 2016.

[29] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Interference mitigation via pricing in time-varying cognitive radio systems," in *NetGCoop '16: Proceedings of the 2016 International Conference on Network Games, Control and Optimization*, 2016.

[30] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online interference mitigation via learning in dynamic IoT environments," in *GLOBECOM '16: Proceedings of the 2016 IEEE Global Telecommunications Conference*, 2016.

[31] A. Marcastel, E. V. Belmega, P. Mertikopoulos, and I. Fijalkow, "Online power allocation for opportunistic radio access in dynamic OFDM networks," in *VTC '16-Fall: Proceedings of the 2016 IEEE Vehicular Technology Conference*, 2016.

[32] E. V. Belmega and P. Mertikopoulos, "Energy-efficient power allocation in dynamic multi-carrier systems," in *VTC '15-Spring: Proceedings of the 2015 IEEE Vehicular Technology Conference*, (Glasgow, Scotland), May 2015.

[33] S. D'Oro, P. Mertikopoulos, A. L. Moustakas, and S. Palazzo, "Cost-efficient power allocation in OFDMA cognitive radio networks," in *EUCNC' 15: Proceedings of the 2015 European Conference on Networks and Communications*, 2015.

[34] I. Stiakogiannakis, P. Mertikopoulos, and C. Touati, "No more tears: A no-regret approach to power control in dynamically varying MIMO networks," in *WiOpt '15: Proceedings of the 13th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2015.

[35] S. D'Oro, P. Mertikopoulos, A. L. Moustakas, and S. Palazzo, "Adaptive transmit policies for cost-efficient power allocation in multi-carrier systems," in *WiOpt '14: Proceedings of the 12th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2014.

[36] P. Coucheney, B. Gaujal, and P. Mertikopoulos, "Distributed optimization in multi-user MIMO systems with imperfect and delayed information," in *ISIT '14: Proceedings of the 2014 IEEE International Symposium on Information Theory*, 2014.

[37] G. Bacci, E. V. Belmega, P. Mertikopoulos, and L. Sanguinetti, "Energy-aware competitive link adaptation in small-cell networks," in *WiOpt '14: Proceedings of the 12th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2014.

[38] I. Stiakogiannakis, P. Mertikopoulos, and C. Touati, "No regrets: Distributed power control under time-varying channels and QoS requirements," in *Allerton '14: Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing*, 2014.

[39] J. Lepping, P. Mertikopoulos, and D. Trystram, "Accelerating population-based search heuristics by adaptive resource allocation," in *GECCO '13: Proceedings of the 15th ACM Annual Conference on Genetic and Evolutionary Computation*, pp. 1165–1172, 2013.

[40] P. Mertikopoulos and E. V. Belmega, "Adaptive spectrum management in MIMO-OFDM cognitive radio: An exponential learning approach," in *ValueTools '13: Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, 2013.

[41] P. Mertikopoulos and A. L. Moustakas, "Entropy-driven optimization dynamics for Gaussian vector multiple access channels," in *ICC '13: Proceedings of the 2013 IEEE International Conference on Communications*, 2013.

[42] P. Mertikopoulos and A. L. Moustakas, "Riemannian-geometric optimization methods for MIMO multiple access channels," in *ISIT '13: Proceedings of the 2013 IEEE International Symposium on Information Theory*, 2013.

[43] P. Mertikopoulos, E. V. Belmega, and A. L. Moustakas, "Matrix exponential learning: Distributed optimization in MIMO systems," in *ISIT '12: Proceedings of the 2012 IEEE International Symposium on Information Theory*, pp. 3028–3032, 2012.

[44] P. Mertikopoulos, "Strange bedfellows: Riemann, Gibbs and vector Gaussian multiple access channels," in *NetGCoop '12: Proceedings of the 2012 International Conference on Network Games, Control and Optimization*, 2012.

[45] P. Mertikopoulos, E. V. Belmega, A. L. Moustakas, and S. Lasaulce, "Dynamic power allocation games in parallel multiple access channels," in *ValueTools '11: Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools*, 2011.

[46] P. Mertikopoulos and A. L. Moustakas, "Selfish routing revisited: Degeneracy, evolution and stochastic fluctuations," in *ValueTools '11: Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools*, 2011.

[47] P. Kazakopoulos, P. Mertikopoulos, A. L. Moustakas, and G. Caire, "Distribution of MIMO mutual information: a large deviations approach," in *ITW '09: Proceedings of the 2009 IEEE Information Theory Workshop*, 2009.

[48] P. Mertikopoulos and A. L. Moustakas, "Learning in the presence of noise," in *GameNets '09: Proceedings of the 1st International Conference on Game Theory for Networks*, 2009.

[49] P. Mertikopoulos, A. L. Moustakas, and N. Dimitriou, "Vertical handover between wireless service providers," in *WiOpt '08: Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2008.

[50] N. Dimitriou, P. Mertikopoulos, and A. L. Moustakas, "Vertical handover between wireless standards," in *ICC '08: Proceedings of the 2008 IEEE International Conference on Communications*, 2008.

[51] P. Mertikopoulos and A. L. Moustakas, "The simplex game: Can selfish users learn to operate efficiently in wireless networks?," in *ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, 2007.

SOFTWARE (1)

[1] P. Mertikopoulos, "`GameSeer`: visualization software for game dynamics." Available under the GNU public license at: `http://mescal.imag.fr/membres/panayotis.mertikopoulos/files/GameSeer.zip`, 2014.

DISSERTATIONS (2)

[1] P. Mertikopoulos, *Stochastic perturbations in game theory and applications to networks*. PhD thesis, National and Kapodistrian University of Athens, November 2010.

[2] P. Mertikopoulos, "Gauss's law and residue calculus in the framework of de Rham cohomology," Master's thesis, National and Kapodistrian University of Athens, May 2003.

COLOPHON

This manuscript was typeset with LaTeX 2ε using Robert Slimbach's beautiful *Minion Pro* typeface. The monospaced text (hyperlinks, etc.) was typeset in *Bera Mono*, originally developed by Bitstream, Inc. as "Bitstream Vera" (with Type 1 PostScript fonts by Malte Rosenau and Ulrich Dirr).

The typographic style of this dissertation was inspired by the authoritative genius of Bringhurst's *Elements of Typographic Style*, ported to LaTeX by André Miede, the original designer of the `classicthesis` template. Any unsightly deviations from these works should be attributed solely to the author's (not always successful) efforts to conform to the awkward A4 paper size.

Online Optimization and Learning in Games: Theory and Applications

Grenoble, December 20, 2019

Panayotis Mertikopoulos