University of California
Santa Barbara

Minmax Optimization: New Applications and Algorithms

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Raphael Chinchilla

Committee in charge:

Professor João P. Hespanha, Chair
Professor James B. Rawlings
Professor Upamanyu Madhow
Professor Ramtin Pedarsani

June 2023

The Dissertation of Raphael Chinchilla is approved.

Professor James B. Rawlings

Professor Upamanyu Madhow

Professor Ramtin Pedarsani

Professor João P. Hespanha, Committee Chair

November 2022

Minmax Optimization:

New Applications and Algorithms

by

Raphael Chinchilla

To my family and friends

⋮

and to whomever takes their time to read this dissertation

Acknowledgements

My decision to do a Ph.D. was made about 20 years ago, when I discovered one could be called a doctor without having to go to med school. To be fair, I have always liked to (over) explain things, and always preferred theory to practice, so academia sounded perfect. Paraphrasing my best-men's speech from my wedding, one should not expect less from a 16 year old teenager who likes to wear clothing with elbow-patches.

And yet, even with all the stars aligned, it still was often challenging. There were several moments of self doubt, thinking I was an impostor. Moments of frustration when I could not prove a theorem or get some code to run. And moments of research alone, as about half of my Ph.D. was done remotely, due to Covid-19. All of that to say, the last five years could have been terrible if it had not been for the people around me that made the worst moments tolerable, and the best moments amazing.

Obviously, I have to start by thanking Prof. Hespanha for his mentorship. While I was doing my masters in Paris, I heard terrible stories about advisors, and I was very eager to find an advisor that struck the balance between being demanding and understanding. I do not think I could have chosen better. There was no moment over the last five years in which I did not find in João someone willing to give advice and to hear me. He allowed me to explore topics without asking for any immediate return, and I will be ever thankful for his approach. This dissertation would have been completely different if I had not had the opportunity to do so.

I would also like to acknowledge the faculty at the Center for Control, Dynamical Systems and Computation for their dedication to teaching high-quality courses and for inviting speakers every week to present new research to us. A special acknowledgment goes to the members of my committee, Profs. Madhow, Pedarsani and Rawlings for their constructive feedback. I would also like to acknowledge Prof. Teel, who has given me much advice over the years. I would also like to express my gratitude to the staff of the ECE department, in particular to Val de Veyra and Stephanie Löfgren for their constant support to me and grad students in general.

My friends and colleagues were also fundamental for my success. Thank you labmates Henrique, Sharad, Murat, Guosong, Sean, Matt, Madeline, Pedro and Justin for always be willing to go to the white board and have a discussion about research (or to be frank, have a discussion about anything). Also, thank you for all of those fries-days which was always the best way to finish the week. Thank you to my other friends at UCSB, Liz, Simone, Amanda, Danny, Matina and Grace for making me and the people in the lab stop talking about our research. For my friends from Brasil whom I have missed, even though they were not present with me, they were still in my heart. While there are too many of them to mention, I have a particular gratitude to Veronica, Marie, Rodrigo, Lia, Thais, Chamma, Beck, Lilian, Flora and Joana as well as to my friends in the groups, Oh Drama, Feudo do Wilson, Breja da Revolução and A Turma mais Formada. My time at UCSB was also one of community engagement. I thank the people at the Graduate Student Association for their defense of grad students' interests, in particular Heather, Cierra and Pedro for their lasting friendship beyond our work. I also thank the people in the board of directors of the Isla Vista Food Co-op for their dedication to provide affordable and healthy food for the community, especially Lisa, Jillian and Kent.

My family has always been supportive. Thank you mamãe and papa for the love you gave me, as well as the education and the opportunities to follow my dreams. Titó e Soso, thank you for being such good siblings and friends, and for always pushing me out of my comfort zone. Thank you vovô, vovó and manina for the unlimited love you gave to me and to my parents. Tania, Lau and Edu, I love that I am actually close to you all, you are amazing aunt and cousins. Thank you Kathy, Liv, Ethan, Shug and Niko for having been my family in California; you immediately adopted me and I am ever so grateful for that. To all of you, I am sorry I was never able to explain to you what I do.

Finally, thank you Rachel. I met you two weeks into my Ph.D., literally the day classes started. You have been perfect during these years. Thank you for understanding the commitments of grad school, for understanding when I had to work late, over the weekends, or on the beach while you sun bathed. Thank you for hearing me explain to you the details of what I had been working on. I don't know whether you ever actually understood what is gradient descent, Newton's method or minmax, but you were always willing to hear me talking about it for hours. Of all the people, you are the one who has seen the most my ups and downs over the last years, and you were always there to console me when needed, and to celebrate when it was deserved.

A last and more distant acknowledgment goes to Issac Newton. Thank you for coming up with calculus and for inventing your root finding method. Thank you for being the giant on whose shoulder I ultimately stand.

per aspera ad astra

Curriculum Vitæ
Raphael Chinchilla

Education

2022	Ph.D. in Electrical and Computer Engineering (Expected), University of California, Santa Barbara.
2021	M.Sc. in Electrical and Computer Engineering, University of California, Santa Barbara.
2016	M.Sc. in Electrical Engineering, Télécom Paris.
2016	M.Sc. in Electrical Engineering, University of Paris-Saclay.
2016	B.Sc. in Electrical Engineering, University of São Paulo

Experience

2017-2022	Graduate Student Researcher, Center for Control, Dynamical Systems and Computation at the University of California, Santa Barbara
2022	Summer Intern, Mitsubishi Electric Research Laboratories
2020	Summer Intern, Mitsubishi Electric Research Laboratories
2016	Graduate Student Researcher, Laboratoire des Signaux et Systèmes at the French Centre National de la Recherche Scientifique
2013-2014	Undergraduate Student Researcher, Laboratório de Automação Agrícola, University of São Paulo

Publications

R. Chinchilla, G. Yang, and J. P. Hespanha, "Newton and interior-point methods for (constrained) nonconvex-nonconcave minmax optimization with stability guarantees." Submitted for journal publication, May, 2022

R. Chinchilla and J. P. Hespanha, *Stochastic programming using expected value bounds*, *Transactions on Automatic Control*. Submitted Dec. 2020, Accepted June. 2022. To appear

J. P. Hespanha, R. Chinchilla, R. R. Costa, M. K. Erdal, and G. Yang, *Forecasting COVID-19 cases based on a parameter-varying stochastic SIR model*, *Annual Reviews in Control,* Pandemic Special Issue (2021)

R. Chinchilla, G. Yang, M. K. Erdal, R. R. Costa, and J. P. Hespanha, *A Tale of Two Doses: Model Identification and Optimal Vaccination for COVID-19*, in *Proc. of the 60th IEEE Conf. on Decision and Contr.*, Dec., 2021

R. Chinchilla and J. P. Hespanha, *Optimization-based estimation of expected values with application to stochastic programming*, in *Proc. of the 58th IEEE Conf. on Decision and Contr.*, Dec., 2019

F. Orieux and R. Chinchilla, *Semi-unsupervised Bayesian convex image restoration with location mixture of Gaussian*, in *Proc. of the 25th IEEE European Signal Processing Conference*, Aug., 2017

F. Orieux and R. Chinchilla, *Restauration d'image par une approche bayesienne semi non supervisee et le melange de gaussienne*, in *26eme Colloque GRETSI Traitement du Signal et des Images, GRETSI 2017*, 2017

W. F. Costa, M. J. Bieleveld, R. G. Chinchilla, and A. M. Saraiva, *Segmentation of land use maps for precision agriculture*, in *Anais do VIII Workshop de Computacao Aplicada a Gestao do Meio Ambiente e Recursos Naturais*, SBC, 2017

Abstract

Minmax Optimization:

New Applications and Algorithms

by

Raphael Chinchilla

Minmax optimization is a powerful framework to model optimization problems for which there is uncertainty with respect to some parameters. In this dissertation we look at new applications of minmax optimization and at new algorithms to solve the optimizations.

The new applications revolve around a connection we have developed between minmax optimization and the problem of minimizing an expected value with stochastic constraints, known in the literature as stochastic programming. Our approach is based on obtaining sub-optimal solutions to the stochastic program by optimizing bounds for the expected value that are obtained by solving a deterministic minmax optimization problem that uses the probability density function to penalize unlikely values for the random variables. We illustrate this approach in the context of three applications: finite horizon optimal stochastic control, with state or output feedback; parameter estimation with latent variables; and nonlinear Bayesian experiment design.

As for new algorithms, they are aimed at addressing the problem of finding a local solution to a nonconvex-nonconcave minmax optimization. We propose two main algorithms. The first category of algorithms are modified Newton methods for unconstrained and constrained minmax optimization. Our main contribution is to modify the Hessian matrix of these methods such that, at each step, the modified Newton update direction can be seen as the solution to a quadratic program that locally approximates the minmax problem. Moreover, we show that by selecting the modification in an appropriate way, the only stable points of the algorithm's iterations are local minmax points. The second category of algorithms are a variation of the learning with opponent learning awareness (LOLA) method, which we call Full LOLA. The rationale of (Full) LOLA is to construct algorithms such that the minimizer and maximizer choose their update direction taking into account the response of their adversary to their choice. The relation between our method and LOLA is that the latter can be seen as a first order linearization of the Full LOLA. We show that it is possible to establish asymptotic convergence results for our method both using fix step length and a variation of the Armijo rule.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimization is a ubiquitous topic, present in one form or another in any quantitative field of knowledge. The idea of finding a better (or the best) way to do something is quite natural to humans, and we use it almost daily without necessarily noticing, in questions like "what is the fastest way to drive to work?", "how can I be a better friend?", or even "how can I do the best PhD while maintaining a good work-life balance?" (the last question is an example of unfeasible optimization).

This dissertation looks at the problem of optimization with uncertainties. Consider the example "what is the fastest way to drive to work?" from above. The time that one takes will, in general, be impacted not only by distance and speed limits, but also by traffic. The latter is fundamentally uncertain, so the choice of route that will be taken will generally depend on how one wants to take into account this uncertainty in their decision making process. There are two fundamental ways to treat uncertainties. In the first one, the uncertainty is treated as a random variable, and the goal is to optimize the average cost. Using the same example, this would mean we answer the question "what is the fastest way **on average** to drive to work?". This approach is called stochastic programming. For the second approach, the uncertainty is treated as an adversarial variable, meaning we want to minimize the worst possible cost. Using the same example, this would mean we answer the question "what is the fastest way to drive to work **given the worst possible traffic**". This approach is called minmax optimization (also known as robust optimization).

The first contribution of this dissertation, in Chapter 2, is to show the existence of a relationship between the stochastic programming problem and a minmax optimization problem. More specifically, we show that it is possible to use minmax optimization to obtain upper and lower bounds on the solution to the stochastic programming problem. This is relevant because most methods to solve stochastic programming tend to be extremely slow. The second contribution of this dissertation, in Chapters 3 and 4 is to propose methods to solve the minmax optimization when the optimization variables take continuous values and the cost function and constraints are differentiable. In Chapter 3 the methods we propose are second order methods, meaning that they are based on the ideas of Newton's root finding method. In Chapter 4, the methods are based on the idea of minimizer and maximizer updating their choice of variable while taking into account the answer of their adversary's action.

The contributions of each chapter are summarized in the sections bellow.

Stochastic Programming Using Expected Value Bounds (Chapter 2)

Optimization of an expected value, also called stochastic programming, appears in countless areas of applied probability and engineering. In optimal stochastic control, a dynamical system is subject to stochastic disturbances, and one wants to find the control that minimizes the expected value of the trajectory tracking error. In maximum likelihood estimation, unobserved variables may need to be integrated out through an expected value, to obtain the likelihood of the observed variables. In machine learning, training a neural network means finding the weights that best classify the expected value of a random variable.

Given a scalar function $V(\cdot)$ and a random vector D, the expected value of $V(D)$ can be lower and upper bounded, respectively, by the minimum and maximum values that $V(\cdot)$ takes over the support of D. Our first results shows how these very crude bounds can be improved by including information encoded in the probability density function (*pdf*) of D. In essence, we solve an optimization over the support of D that includes terms that penalize unlikely realizations for D. This means that we need to compute and solve optimality conditions — and therefore essentially compute derivatives and solve algebraic equations — rather than compute integrals.

The results aforementioned actually define a family of bounds. Two instances of this family, which we call the additive and multiplicative bounds, are particularly useful. The first is more appropriate to problems where the cost function $V(\cdot)$ is polynomial, while the second one is more appropriate when the cost function is exponential. Both the additive and multiplicative bounds are parameterized by a scalar parameter ϵ, which can itself be optimized. To guide the design of the bounds and select ϵ, we develop necessary and sufficient conditions with respect to ϵ that can be used to make sure that the additive and multiplicative bounds are finite.

Borrowing ideas from robust optimization, the bounds are used to compute approximate solutions to stochastic programming optimizations: Instead of minimizing an expected value subject to stochastic constraints, we minimize upper/lower bounds for the criterion subject to constraints on pessimistic/optimistic bounds for the stochastic constraints. For the lower bound, this leads to a minimization on an extended variable space; for the upper bound, it leads to a minmax problem.

Finally, we discuss three applications for our bounds. The first relates to finite-horizon stochastic optimal control, with either state feedback or output feedback. In the former case, the initial state is assumed known, but an expectation is needed over the realization of future disturbances. In the latter case, the initial state is unknown, and the expectation is taken with respect to a conditional distribution, given known realizations of past noisy measurements. Our approach can include stochastic constraints on the trajectory of the system, which we illustrate through a constraint on the final state.

The second application is related to Maximum Likelihood or Maximum a Posteriori estimation involving latent variables that cannot be measured [9]. These problems require the latent variables to be marginalized by an expectation that can be upper/lower bounded using the results from Section 2.2.

The third application is in the area of Bayesian experiment design [10–12]. The goal is to optimize the values of experimental parameters to facilitate the estimation of unknown variables. Experiment design criteria typically involve taking expectations with respect to unknown variables, including the ones that need to be estimated. Also here, optimal experiment design can be performed by replacing expectations by bounds.

In the context of feedback control, all three applications discussed above typically need to be performed in real-time with limited computation, and benefit from the availability of bounds on how the approximate solution compares with the true optimum. It is in such scenarios that the approach proposed here is most attractive. In contrast, when computation is unlimited, Monte Carlo based methods can achieve arbitrarily accurate solutions to stochastic optimization problems as long as one uses a sufficiently large number of samples, and will thus eventually out-perform in accuracy the approach proposed here.

Related Work Stochastic programming has been an active area of research for the last 60 years, therefore a complete overview of the literature is infeasible. Instead, we provide a brief overview of the most fundamental methods, some recent developments and how these relate to our work. We discuss separately four approaches: deterministic methods, stochastic methods, methods based on robust optimization and distributionally robust optimization .

Deterministic methods rely on computing the expected value using a numerical integration method such as Gauss-Kronrod [13, 14]. Since numerical integration is computationally infeasible for large problems, deterministic approximations of the expected value are often used. Common approximations include minimizing the truncated Taylor expansion of the expected value [15–17] and the Laplace and saddle-point approximations [18, 19]. A weakness of these methods is that they generally do not provide guarantees regarding how the solution found compares to the true optimum.

Stochastic optimization methods rely on some form of Monte Carlo sampling. These methods generally scale well and provide confidence intervals on the solutions. The most intuitive method is the Sample Average Approximation (SAA) (also known as Empirical Risk Minimization) [20–22], where the expected value is approximated by the empirical average obtained through sampling. Stochastic Gradient Descent (SGD) [10] is an easy to implementation and versatile alternative. The core idea of SGD is to directly draw samples of the gradient of the expected values, rather than using the gradient of the empirical mean to do gradient descent. Under appropriate assumptions, both SAA and SGD are guaranteed to converge to a (possibly local) minimum as the number of sample grows [22], but accurate results may require a very large number of samples, making these methods not suitable for real time applications.

The Scenario Approach approximately solves chance-constrained optimizations by sampling the constraints [23–25]. This approach guarantees constraint satisfaction with high probability. While the number of samples increases only logarithmically with the confidence parameter (usually denoted by β), it is also proportional to the dimensions of the optimization variables and inversely proportional to the risk parameter (usually denoted by ϵ). As a result, Scenario Approach may require many samples which can lead to high computational complexity. Moreover, while tight requirements on the confidence parameter β have a moderate impact in the number of samples, it typically also lead to more conservative results.

In robust optimization one minimizes for the worst possible perturbation, while guaranteeing some base level of performance [26, 27]. Robust optimization has been gaining popularity in recent years, for examples in fields such as Model Predictive Control [28] and Machine Learning [29–31]. Some new developments have also been made in numerical aspects, notably in [32], where the authors provide first and second order optimality conditions for minmax when the criteria is nonconvex on the minimization variable and nonconcave on the maximization variable. Robust optimization was traditionally not regarded as an approach to solve stochastic programming problems, but in the last decade some articles have connected the two areas, for instance [33–35].

At the intersection between robust and stochastic optimization lies distributionally robust optimization (DRO), where the objective is to minimize an expected value for the worst probability distribution within a set of admissible distributions. This set, called the ambiguity set, is often constructed from samples of the true distribution and its selection tries to balance between expressiveness (how rich is the information in the set) and tractability (how easy it is to solve the DRO). In [36, 37] the authors use the Wasserstein metric to construct the ambiguity set and show that for some classes of problem, the complexity of solving the associated DRO is similar to that of Sample Average Approximation. In [38–41] the ambiguity sets are constructed using sample statistics, such as mean, covariance and entropy. The DRO is reformulated into a minimization on a larger set of variables using tools from duality theory and convex optimization. When the ambiguity set is constructed to guarantee (with high probability) that it contains the true distribution, DRO can also be seen as a method to bound the true expected values, which can be used to solve stochastic programs. For a broader exposition on DRO we refer to [42] and the references within.

Newton and interior-point methods for (constrained) nonconvex-nonconcave minmax optimization with stability guarantees (Chapter 3)

In minmax optimization, one minimizes a cost function which is itself obtained from the maximization of an objective function. Minmax optimization is a powerful modeling framework, generally used to guarantee robustness to an adversarial parameter such as accounting for disturbances in model predictive control [43, 44], security related problems [45, 46], or training neural networks to be robust to adversarial attacks [31]. It can also be used as a framework to model more general problem such as sampling from unknown distributions using generative adversarial networks [47], reformulating stochastic programming as minmax optimization [2, 48, 49], or producing robustness of a stochastic program with respect to the probability distribution [42]. Minmax optimization is also known as minimax or robust optimization. A generalization of minmax optimization is bilevel optimization, where each player has different cost functions [50, 51].

Finding a global minmax point for nonconvex-nonconcave problems is generally difficult, and one has to settle for finding a local minmax point. Surprisingly, only recently a first definition of unconstrained local minmax was proposed in [52], and the definition of constrained local minmax in [53].

The foundation of our work involves examining Newton's method iterations through the lens of dynamical systems. By analyzing the linearization of the dynamics, we deduce that every equilibrium point (i.e., a point with a zero gradient) is locally asymptotically stable. This poses a problem when using Newton's method for nonconvex minimization, as the algorithm is drawn to any equilibrium point, regardless of whether it is a local minimum. Our initial contribution is an examination of the local convergence properties of a modified Newton's method, employing a widely accepted Hessian matrix modification. This entails adding a matrix, typically a scaled identity matrix, to make the modified Hessian positive definite [54, Chapter 3.4]. We demonstrate that incorporating this additive matrix renders every non-local-minimum equilibrium point unstable while maintaining stability for local minima. This is crucial for nonconvex optimization, as it ensures that the Newton iterations can only converge to an equilibrium point if it is a local minimum. Utilizing analogous techniques, we establish similar results for primal-dual interior-point methods in constrained minimization. These findings directly inspire the development of new Newton-type algorithms for minmax optimization.

The modified Newton method presented in the previous paragraph can be viewed as a sequence of local quadratic approximations of the minimization problem. Motivated by this, we develop Newton-type algorithms for minmax optimization, conceptualized as a series of local quadratic approximations of the minmax problem. For convex-concave functions, this quadratic approximation is just the second-order Taylor expansion, which leads to the (unmodified) Newton's method, accompanied by its well-established local convergence properties. However, for nonconvex-nonconcave functions, it is necessary to add scaled identity matrices to ensure that the local approximations possess finite minmax solutions (without mandating convex-concavity). Additive terms meeting this criterion are said to satisfy the Local Quadratic Approximation Condition (LQAC). Employing a sequence of local quadratic approximations acts as a surrogate for guiding the modified Newton's method towards a solution at each step. Nevertheless, we demonstrate that, unlike minimization, local quadratic approximation-based modifications are not enough to ensure that the algorithm can only converge towards local minmax points. Our minmax findings reveal that additional conditions are required on the modification to unsure the algorithm's convergence to an equilibrium point is guaranteed only if that point is a local minmax.

The conditions described above to establish the equivalence between local minmax and local asymptotic stability of the equilibria to a Newton-type iteration are directly used to construct a numerical algorithm to find local minmax. By construction, when this algorithm converges to an equilibrium point, its is guaranteed to obtain a local minmax. One could be tempted to think that the issue of getting instability for the equilibria that are not local minima or that are not local minmax is just a mathematical curiosity, which in practice makes little difference. However, our numerical examples show otherwise. Most especially the pursuit-evasion MPC problem, finding a local minmax (rather than an equilibrium that is not local minmax) leads to a completely different control. Specifically, if the instability property is not guaranteed, the evader is not able escape from the pursuer. It is important to emphasize that our results fall shy of guaranteeing *global* asymptotic convergence to a local minmax, as the algorithm could simply never converge. However, our numerical examples also show that our algorithm seems to enjoy good global convergence properties in practice. Using the results of this paper, we have created a solver for minmax optimization and included it in the solvers of `TensCalc`[1] [55]; this solver was used to generate the numerical results we present.

Related Work Traditionally, robust optimization focused on the convex-concave case, with three main methods. The first type of method is based on Von Neuman's minmax

[1] `https://github.com/hespanha/tenscalc`

theorem [56] that states that the min and the max commute when the problem is convex-concave and the optimization sets are convex and compact. Solving the minmax then simplifies to finding a point that satisfies the first order condition. While there are many different methods to achieve this, many of them can be summarized by the problem of finding the zeros of a monotone operator [57]. The second type of methods consists on reformulating the minmax as a minimization problem which has the same solution as the original problem. This is generally done using either robust reformulation through duality theory or tractable variational inequalities [50, 58–60]. The third, cutting-set methods, solves a sequence of minimization where the constraint of each minimization is based on subdividing the inner maximization [61]. The robust reformulation is problem specific, while the cutting-set approach requires solving many exact maximization which might not be feasible in large scale.

Motivated by some of the shortcomings of these methods and the necessities of machine learning, research on minmax optimization started to study first-order methods based on variations of gradient descent-ascent. The results tend to focus on providing convergence complexity given different convexity/concavity assumptions on the target function. In multi-step gradient descent ascent, also know as unrolled or GDmax, the minimizer is updated by a single gradient descent whereas the maximizer is updated by several gradient ascent steps that aim to approximately find the maximum. In single step, the minimizer and maximizer are updated at each iteration. Variations of the standard gradient descent-ascent includes augmenting it with some distinct features such as different step sizes, or using momentum. A third option, which is completely different from what is described for other methods in this section, is to include the gradient from different time steps in the computation, such as the past one (as in optimistic gradient descent-ascent), the midpoint between the current and future points (as in extra gradient descent-ascent) and at future point (as in proximal point). The literature on first-order methods is very extensive, and we refer to [52, 62–69] and the references within for the exposition on some of these methods and their convergence properties.

In recent years, researchers have also started to work on algorithms that use second order derivatives to determine the directions. These algorithm, in their major part have not attracted as much attention as first order methods. In the Learning with Opponent Learning Awareness (LOLA), the minimizer anticipates the play of the maximizer using the Jacobian of the maximizer's gradient [70, 71]. In competitive gradient descent, both minimizer and maximizer use the cross derivative of the Hessian to compute their direction [72]. In follow the ridge, the gradient ascent step is corrected by a term that avoids a drift away from local maxima [73]. In the total gradient descent-ascent, similarly to LOLA, the descent direction is computed by taking to total derivative of a function which anticipates the maximizer's response to the minimizer [74]. Finally, the complete Newton borrows ideas from follow the ridge and total gradient to obtain a Newton method which prioritizes steps towards local minmax [75]. These three last algorithms are shown to only converge towards local minmax under some conditions, but in none of them it is addressed the issue of how to adjust the Hessian far away from a local minmax point.

Recently, some second order methods have been proposed for the nonconvex-strongly-concave case, where the Hessian is modified such that it is invertible and that the minimizer update is a descent direction of the objective function at its maximum. They either use cubic regularization [76, 77] or randomly perturb the Hessian [78]. Because of some of the assumptions these work make, most important the strong-concavity of the objective function with respect to the maximizer, they are able to establish complexity analysis and guarantee. It is also worth mention that these algorithms are all multi-step based, meaning they (approximately) solve the maximization between each update of the minimizer, whereas our algorithm updates both the minimizer and the maximizer simultaneously.

On the Asymptotic Convergence of Full LOLA (Chapter 4)

Solving a minmax problem, also known as robust optimization, consists on finding the optimal strategies for two players that want to optimize opposite interests. Conceptually, this is a versatile paradigm that can be used to model a variety of situations, including games such chess, elections between two candidates, an airplane flying in the middle of a storm and neural network accurately classifying misleading information. Most modern algorithms to solve minmax problems have players choosing locally their strategy without taking into account what the other player will do. Our goal in this chapter is to develop an algorithm in which each player takes chooses their local strategy while taking into account what will be other player's action.

The modern approach to minmax problems was established in the seminal chapter by von Neumann [79] in which he proved that if the problem is convex in the minimization and concave in the maximization, then the min and the max commute, meaning that the order of the players does not matter. This is known as the Minmax Theorem and has since been extended to other cases [80, 81].

However, in many problems of interest, the min and max do not necessarily commute. Some of these include adversarial learning [31, 82–84], generative adversarial networks [47], robust model predictive control [44, 85, 86], robust estimation [87, 88], robust optimization for stochastic optimization [24, 49, 89], among many.

In some cases, it is possible to solve the non-commuting minmax problems using approaches such as robust counterpart or cutting-set methods [58, 60, 90, 91]. In the other cases, generally when the problem is non-convex non-concave, one is usually restricted to finding local minmax points, as defined in [32], which have first and second order necessary and sufficient conditions obtained from the gradient and Hessian.

An elegant method to look for points satisfying the first order necessary condition is the Learning with Opponent Learning Awareness (LOLA) introduced in [92]. The idea of LOLA is that the minimizer chooses its direction based on the predicted direction the maximizer will take. The convergence of a modified version of LOLA was given by the same group of authors [93].

In this chapter, we introduce the Full LOLA algorithm, of which the standard LOLA can be seen as a linearization of. In our opinion, the Full LOLA approach has several elegant properties which motivated us to explore using it. While we did not found any numerical application that could benefit from this approach instead of using Gradient Descent Ascent or other (strictly) first order methods, we believe the intuitions developed in these proofs might end up being useful either for other proofs or in applications we were not aware of.

The main ingredient of our approach is what we call the full descent ascent directions. In essence, in a full descent ascent direction, the minimizer does not decrease the cost function at the current point, but instead decreases the cost function calculated at the next value that the maximizer will take. This choice of directions reflects the asymmetry of minmax games, in which the minimizer has less freedom to chose their action than the maximizer has. Building from this definition, we propose a method to obtain full descent ascent directions based on gradients. For the maximizer, this is equivalent to gradient ascent. For the minimizer, the descent direction is obtained from a modified version of the cost function, in which the value of the maximizer is offset by the gradient ascent step. The method we propose is actually slightly more general, and allow us to solve problems with convex constraints. It also allow us to use scaling matrices to compute the directions, such as the Hessian, obtaining Newton types algorithms. We prove the asymptotic convergence of the method, for two types of step sizes, either fixed or adjusted using an Armijo rule.

Chapter 2

Stochastic Programming Using Expected Value Bounds

Parts of this chapter come from [2].

In this chapter we consider the problem of using minmax optimization as a proxy to solve the problem of minimizing an expected value with stochastic constraints, known in the literature as stochastic programming. Stochastic programming problems are generally solved using sample based approaches such as stochastic gradient descent or sample average approximation. These methods require many samples from the underlying distribution, and are too slow to be used in real time applications. As we show in the remaining chapters of this dissertation, minmax optimization is substantially faster to solve, which allows one to obtain accurate solutions to the stochastic programming problem order of magnitude faster.

We start this chapter by presenting a family of bounds on the expected value of a cost function, in Section 2.1. The fundamental idea is to substitute the expected value by a deterministic maximization over the support of the random variable while also penalizing unlikely variables. This approach allows one to obtain lower and upper bound on any scalar expected value.

Based on these results, in Section 2.2 we present how these bounds can be used in the context of stochastic programming. This is obtained by substituting the expected values in the cost function and in the stochastic constraints by the bounds we proposed in the previous section. We show that this proxy minmax problem can be used to bound the original problem. We also show what are the conditions on the problem such that the bounds can be solved using numerical tools.

Finally, in Section 2.3 we apply our bounds to three type of problems: stochastic control of a discrete time system, estimation in the presence of latent variables, and experiment design. In all of these applications, we show that our algorithm is able to find solutions similar to the ones obtained by a stochastic optimization algorithm, but orders of magnitude faster.

Notation Given an underlying probability space $(\Omega, \mathcal{F}, \mathrm{P})$, a random variable X and a scalar $x \in \mathbb{R}$, we denote by $\mathrm{P}(X \leqslant x)$ the probability measure of the set $\{\omega \in \Omega : X(\omega) \leqslant x\} \in \mathcal{F}$ and by $\mathbb{E}[X]$ the expected value of X. Given a measurable event $E \in \mathcal{F}$ with $\mathrm{P}(E) > 0$, we define conditional essential infimum and supremum by

$$\begin{aligned} \operatorname{ess\,inf}[X \mid E] &= \sup\{x \in \mathbb{R} : \mathrm{P}(X \geqslant x \mid E) = 1\} \\ \operatorname{ess\,sup}[X \mid E] &= \inf\{x \in \mathbb{R} : \mathrm{P}(X \leqslant x \mid E) = 1\}. \end{aligned}$$

Unconditional essential infimum and supremum are denoted simply by $\operatorname{ess\,inf} X$ and $\operatorname{ess\,sup} X$ and correspond to the case $E = \Omega$. The essential supremum and infimum relax the usual supremum and infimum by excluding sets of measure zero. One can informally think of them as $\sup_{x \in \mathcal{X}} x$ and $\inf_{x \in \mathcal{X}} x$ where $\mathcal{X}$ is the support of X.

Given two random variables X, Y we use the notation $X \overset{\text{wpo}}{\leqslant} Y$ when $\mathrm{P}(X \leqslant Y) = 1$ and analogously for $\overset{\text{wpo}}{\geqslant}, \overset{\text{wpo}}{<}, \overset{\text{wpo}}{>}$.

2.1 Bounds on an expected value

Given a random vector D taking values in $\mathcal{D} \subset \mathbb{R}^M$ and a scalar measurable function $V : \mathcal{D} \to \mathbb{R}$, the monotonicity of the expected value $\mathbb{E}[V(D)]$ provides the following basic bound

$$\text{ess inf}\, V(D) \leqslant \mathbb{E}[V(D)] \leqslant \text{ess sup}\, V(D). \tag{2.1}$$

The core idea of this section is to improve upon this crude bound by including information about D, for example, coming from its probability density function (*pdf*). To present our first result, we introduce the following terminology. Consider a right-ordered group $G \coloneqq (\mathcal{P}, \oplus)$ defined on a set $\mathcal{P} \subset \mathbb{R}$ for which the group operation $\oplus$ satisfies the usual group properties of closure, associativity, existence of an identity element, and existence of inverse elements (which we denote using $\neg$); as well as the right-ordered property

$$a \leqslant b \quad \Rightarrow \quad a \oplus c \leqslant b \oplus c, \quad \forall a, b, c \in \mathcal{P}$$

[94]. We say that $G \coloneqq (\mathcal{P}, \oplus)$ is *distributive with respect to integration (or E-distributive for short)* if it is right-ordered and, for every random variable X taking values on $\mathcal{P}$, we have that

$$a \oplus \mathbb{E}[X] = \mathbb{E}[a \oplus X], \quad \forall a \in \mathcal{P}.$$

Theorem 2.1.1 (Bounds on an expected value) *Consider an E-distributive group $G \coloneqq (\mathcal{P}, \oplus)$, a random vector D taking values in $\mathcal{D} \subset \mathbb{R}^M$, and measurable functions $V, \alpha : \mathcal{D} \mapsto \mathcal{P}$. If $\mathbb{E}[V(D)]$ and $\mathbb{E}[\neg\alpha(D)]$ are finite, then*

$$\text{ess inf}\, J(D) \leqslant \mathbb{E}[V(D)] \leqslant \text{ess sup}\, J(D) \tag{2.2}$$

where the function $J : \mathcal{D} \to \mathbb{R}$ is defined by

$$J(d) \coloneqq V(d) \oplus \alpha(d) \oplus \mathbb{E}[\neg\alpha(D)]. \qquad \square$$

Proof. We prove the upper bound, the proof for the lower bound can be obtained analogously. For every scalar $v \geqslant \operatorname{ess\,sup}\ V(D) \oplus \alpha(D)$, we have that

$$\mathrm{P}(V(D) \oplus \alpha(D) \leqslant v) = 1,$$

by the definition of essential supremum. From the monotonicity of the expected value, we thus conclude that

$$\mathbb{E}\big[V(D) \oplus \alpha(D)\big] \leqslant v.$$

Since $\mathbb{E}[\neg\alpha(D)]$ is finite, we can use the right-ordered property of $(\mathcal{P}, \oplus)$ to conclude that

$$\mathbb{E}\big[V(D) \oplus \alpha(D)\big] \oplus \mathbb{E}[\neg\alpha(D)] \leqslant v \oplus \mathbb{E}[\neg\alpha(D)]$$

and then the E-distributed property to obtain

$$\mathbb{E}\big[V(D) \oplus \alpha(D) \oplus \neg\alpha(D)] = \mathbb{E}[V(D)] \leqslant v \oplus \mathbb{E}[\neg\alpha(D)].$$

The upper bound then follows by taking an infimum on the right-hand side over the set of such scalars $v \geqslant \operatorname{ess\,sup}\ V(D) \oplus \alpha(D)$. ■

The key idea of Theorem 2.1.1 is to improve upon (2.1) by including in $J(\cdot)$ terms that reduce the essential supremum and increase the essential infimum. To reduce the supremum, for example, one should select $\alpha(d)$ so that it is strongly negative (in the sense that $\neg\alpha(d)$ should be strongly positive) when $V(d)$ is large and while keeping $\mathbb{E}[\neg\alpha(D)]$ relatively small. In the remainder of the chapter we mostly use two E-distributive groups G and associated functions α that achieve this for our applications of interest. Both bounds assume that D has a probability density function (*pdf*) that we denote by $p_D(\cdot)$.

Additive Bound: The E-distributive group $(\mathcal{P}, \oplus) = (\mathbb{R}, +)$ with the usual addition of reals, and $\alpha(d) = \epsilon \log p_D(d)$ with $\epsilon \in \mathbb{R}$, leads to

$$J(d, \epsilon) \coloneqq V(d) + \epsilon\, \log p_D(d) + \epsilon\, \mathcal{H}_D, \tag{2.3}$$

where $\mathcal{H}_D \coloneqq \mathbb{E}[-\log p_D(D)]$ is the differential entropy.

Multiplicative Bound: The E-distributive group $(\mathcal{P}, \oplus) = (\mathbb{R}_{>0}, \times)$ with the usual multiplication of positive reals, and $\alpha(d) = p_D(d)^\epsilon$ with $\epsilon \in \mathbb{R}$, leads to

$$J(d, \epsilon) \coloneqq V(d)\, p_D(d)^\epsilon\, \mathcal{I}_D(\epsilon) \tag{2.4}$$

where $\mathcal{I}_D(\epsilon) \coloneqq \mathbb{E}[p_D(D)^{-\epsilon}]$.

The functions J in (2.3) and (2.4) are not necessarily well defined on the measure zero set where $p_D(D) = 0$, but the value of J on such set is irrelevant, as it does not affect the value of the essential supremum or infimum in (2.2).

The key idea behind the additive bound is that unlikely values d for D will lead to a large negative value for $\log p_D(d)$ and reduce the value of $J(d)$. These unlikely values will contribute with a strong positive value in $-\log p_D(D)$, but precisely because they are unlikely, they will not increase $\mathcal{H}_D \coloneqq \mathbb{E}[-\log p_D(D)]$ very much. Overall, this should thus decrease the supremum of $J(d)$ over $\mathcal{D}$ to create a tighter bound. A similar reason can be used to justify the function α proposed for the multiplicative bound.

In Appendix 2.A.1, we derive expressions for $\log p_D(d) + \mathcal{H}_D$ and $p_D(d)^\epsilon \mathcal{I}_D(\epsilon)$ for the Gaussian and for the uniform distributions.

Remark 2.1.2 (Bounds for conditional expectation) *Theorem 2.1.1 can also be stated for conditional expectations, provided that the E-distributive property holds for the conditional expectation with probability one. In this case, the additive and multiplicative bounds should involve conditional pdf.* □

2.1.1 Selection of bound and ϵ

It is possible to establish necessary and sufficient conditions such that the additive and multiplicative bounds lead to non-trivial results, which are presented in Appendix 2.A.3.

Here, we present a corollary of those results that includes the sufficient conditions which, in practice, are the most useful in deciding which bounds to use. We require the following definition to present the corollary. Given a constant $\gamma > 0$ sufficiently small so that $\mathrm{P}\big(p_D(D) > \gamma\big) > 0$, we say that a measurable function $f(\cdot)$ is γ-*essentially upper bounded* if

$$\text{ess sup}\big[f(D) \mid p_D(D) > \gamma\big] < \infty,$$

γ-*essentially lower bounded* if

$$\text{ess inf}\big[f(D) \mid p_D(D) > \gamma\big] > -\infty,$$

and γ-*essentially bounded* if it is both γ-essentially upper and lower bounded. γ-essential boundedness is a much milder requirement than the usual notion of boundedness, as it allows functions to become very large (growing all the way to infinity) as long as the pdf becomes sufficiently small.

Corollary 2.1.3 (Sufficient conditions for finite bounds) *Assume that $p_D(\cdot)$ is γ-essentially upper bounded and consider finite constants $\epsilon \in \mathbb{R}$ and $c \in (0, 1/\gamma)$ such that*

$$\text{ess sup}\big[p_D(D) \mid p_D(D) > \gamma\big] \leqslant 1/c.$$

When $V(\cdot)$ is γ-essentially bounded, we have that

$$p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \text{ess inf}\Big[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] > \epsilon$$
$$\Rightarrow \quad \text{ess inf}\big(V(D) + \epsilon \log p_D(D)\big) > -\infty.$$

and

$$p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \text{ess sup}\Big[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] < \epsilon$$

$$\Rightarrow \quad \text{ess sup}\big(V(D) + \epsilon \log p_D(D)\big) < +\infty$$

Alternatively, when $\log V(\cdot)$ *is* γ*-essentially bounded, we have that*

$$p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \text{ess inf}\left[\frac{-\log V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\right] > \epsilon$$
$$\Rightarrow \quad \text{ess inf}\big(\log V(D) + \epsilon \log p_D(D)\big) > -\infty.$$

and

$$p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \text{ess sup}\left[\frac{-\log V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\right] < \epsilon$$
$$\Rightarrow \quad \text{ess sup}\big(\log V(D) + \epsilon \log p_D(D)\big) < +\infty. \qquad \square$$

The first two implications in Corollary 2.1.3 involve $V(D)$ and are relevant for the additive bound, while the remaining ones involve $\log V(D)$ and the multiplicative bound.

Specifically, this result establishes that for the additive and multiplicative bounds to be non trivial (i.e., finite), it suffices to pick an ϵ such that $\log c\, p_D(D)$ dominates either $V(D)$ or $\log V(D)$, respectively. Therefore, which bound to use essentially depends on the rates of growth of $V(\cdot)$, $\log V(\cdot)$, and $\log p_D(\cdot)$. When both bounds have a finite value, we have observed that the approximations seems to be better when $V(\cdot)$ (or $\log V(\cdot)$) has roughly the same magnitude as $\log p_D(\cdot)$.

Among the values of ϵ that lead to a finite upper bound, the conservativeness of the bound can be minimized by selecting the value of $\epsilon \in \mathbb{R}$ that minimizes

$$\inf_{\epsilon \in \mathbb{R}} J^*(\epsilon), \quad J^*(\epsilon) \coloneqq \text{ess sup}\, J(D, \epsilon) \tag{2.5}$$

with $J(d, \epsilon)$ as in (2.3) or (2.4). It turns out that such minimization over the scalar parameter ϵ is well-behaved as the function $J^*(\epsilon)$ in (2.5) has appropriate convexity properties, as noted in the following result proved in Appendix 2.A.2:

Proposition 2.1.4 (Optimization over ϵ) *The function $J^*(\epsilon)$ in* (2.5) *is convex for $J(d, \epsilon)$ in* (2.3) *and log-convex for $J(d, \epsilon)$ in* (2.4)*. Moreover, $J^*(\epsilon)$ is finite on a convex set.* □

Remark 2.1.5 (Beyond the additive and multiplicative bounds) *Most of the discussion in this section and the application examples discussed in Section 2.3 make use of the additive and multiplicative bounds. However, these do not necessarily provide the tightest bounds. Consider for example a chi-square random variable D with 1 degree of freedom, whose pdf is given by $p_{\chi^2}(d) = \frac{e^{-\frac{d}{2}}}{\sqrt{2\pi d}}$, $\forall d > 0$ and is known to have an expected value $\mathbb{E}[D]$ equal to 1. The additive upper bound from* (2.3) *is not useful as it leads to* $\forall \epsilon \in \mathbb{R}$

$$\sup_{d>0} \left(d + \epsilon \log \left(\frac{e^{-\frac{d}{2}}}{\sqrt{2\pi d}} \right) \right) + \epsilon \mathcal{H}_{\chi^2} = +\infty,$$

where $\mathcal{H}_{\chi^2}$ is the entropy of D. In contrast, the multiplicative upper bound from (2.4) *leads to the following finite bound*

$$\begin{aligned} &\inf_{\epsilon \in \mathbb{R}} \sup_{d>0} d \left(\frac{e^{-\frac{d}{2}}}{\sqrt{2\pi d}} \right)^{\epsilon} \left(\frac{1}{2\pi} \right)^{\frac{1-\epsilon}{2}} \left(\frac{2}{1-\epsilon} \right)^{\frac{1+\epsilon}{2}} \Gamma\left(\frac{1+\epsilon}{2} \right) \\ &\approx 1.478. \end{aligned}$$

However, a tight bound can be obtained using the multiplicative group $(\mathcal{P}, \oplus) = (\mathbb{R}_{>0}, \times)$ together with the function $\alpha(d) = d^{-\epsilon}$, which leads to

$$\inf_{\epsilon \in \mathbb{R}} \sup_{d \geqslant 0} \left(d\, d^{-\epsilon} \frac{2^{\epsilon}\, \Gamma(0.5 + \epsilon)}{\Gamma(0.5)} \right) = 1.$$

While either the additive or the multiplicative bound typically lead to reasonable bounds, this example shows that it may be worth it to explore alternatives. □

Remark 2.1.6 (Unknown pdf) *When the pdf $p_D(\cdot)$ of D is not explicitly known, it is not easy to use the additive and multiplicative bounds in* (2.3)*–*(2.4)*, because both include*

$p_D(d)$ in the criteria to be optimized over $d \in \mathcal{D}$. In such cases, one can still use the bounds in Theorem 2.1.1, but with functions $\alpha(d)$ that do not explicitly include the pdf of D. We recall from the discussion right after Theorem 2.1.1, that the key to get a tight upper bound is to select for $\alpha(d)$ a function that is strongly "negative" when $V(d)$ is large, and yet $\mathbb{E}[\neg\alpha(D)]$ is relatively small. For the additive group, the function $\alpha(d) \coloneqq \log p_D(d)$ typically has this property when large values for $V(d)$ have low probability. When the pdf is unknown, tight bounds can still be obtained as long as one selected for $\alpha(d)$ values that are strongly negative when $V(d)$ is large and yet $D = d$ is unlikely. □

2.1.2 Connection to distributionally robust optimization

Distributionally robust optimization (DRO) can provide an alternative approach to compute bounds for an expected value by noting that

$$\inf_{\bar{\mathrm{P}} \in \mathcal{P}} \mathbb{E}_{\bar{\mathrm{P}}}[V(D)] \leqslant \mathbb{E}_{\mathrm{P}}[V(D)] \leqslant \sup_{\bar{\mathrm{P}} \in \mathcal{P}} \mathbb{E}_{\bar{\mathrm{P}}}[V(D)], \tag{2.6}$$

where the subscript in the expected value operator refers to the probability measure used for the computation of the expected value and $\mathcal{P}$ denotes some class of probability measures that contains the actual measure P. From a computational perspective, such bounds can be useful when the minimum and maximum over $\mathcal{P}$ are achieved for measures $\bar{P}$ for which the expectation $\mathbb{E}_{\bar{\mathrm{P}}}[V(D)]$ is easier to compute than the original $\mathbb{E}_{\mathrm{P}}[V(D)]$. For example, if we include in $\mathcal{P}$ every distribution for which D is measurable, we essentially get the trivial bounds in (2.2).

It turns out that (2.6) can lead to bounds closely related to those obtained in Theorem 2.1.1: Suppose for simplicity that we focus our attention on a discrete random variable $D \in \{d_1, d_2, \ldots, d_K\}$ and pick for $\mathcal{P}$ the set of all distributions with entropy larger than or equal to the entropy $\mathcal{H}[\mathrm{P}]$ of the actual probability distribution P. In this case, the upper bound in (2.6) is of the form

$$\mathbb{E}_{\mathrm{P}}[V(D)] \leqslant \max_{\bar{p}_1,\ldots,\bar{p}_K} \Big\{ \sum_{k=1}^{K} V(d_k)\bar{p}_k : -\sum_{k=1}^{K} \log(\bar{p}_k)\bar{p}_k \geqslant \mathcal{H}[\mathrm{P}] \Big\}, \tag{2.7}$$

where the maximization is taken over the simplex of probability distributions. Because the entropy is a strictly concave function, as long as P is not the uniform distribution, $\bar{p}_k = 1/K, k \in \{1, \ldots, K\}$ is a Slater point and strong duality holds, which allow us to replace the right-hand side of (2.7) by its dual problem:

$$\mathbb{E}_{\mathrm{P}}[V(D)] \leqslant \inf_{\epsilon \leqslant 0} \max_{\bar{p}_1,\ldots,\bar{p}_K} \sum_{k=1}^{K} V(d_K)\bar{p}_k + \epsilon \sum_{n=1}^{K} \log(\bar{p}_k)\bar{p}_k + \epsilon \mathcal{H}[\mathrm{P}]. \tag{2.8}$$

For the same expected value, the additive upper bound provided by Theorem 2.1.1 is of the form

$$\mathbb{E}_{\mathrm{P}}[V(D)] \leqslant \inf_{\epsilon \in \mathbb{R}} \max_{k} V(d_k) + \epsilon \log(p_k) + \epsilon \mathcal{H}[\mathrm{P}], \tag{2.9}$$

where, as in (2.5), we pick the least conservative upper bound over the range of parameters $\epsilon \in \mathbb{R}$. It turns out that the maximum over k in (2.9) has the same numerical value as the following maximization over the simplex of distributions:

$$\mathbb{E}_{\mathrm{P}}[V(D)] \leqslant \inf_{\epsilon \in \mathbb{R}} \max_{\bar{p}_1,\ldots,\bar{p}_K} \sum_{n=1}^{K} V(d_k)\bar{p}_k + \epsilon \sum_{k=1}^{K} \log(p_k)\bar{p}_k + \epsilon \mathcal{H}[\mathrm{P}], \tag{2.10}$$

leading to a bound strikingly similar to (2.8). However, the two bounds generally lead to different numerical values:

i) For distributions with large entropy, the DRO inequalities (2.7) and (2.8) lead to a tighter bound, because the family of distributions that satisfy the constraint in (2.7) becomes fairly small. In fact, for the uniform distribution $p_k = 1/K$, $\forall k$ with maximal entropy $\mathcal{H}[\mathrm{P}] = \log K$, only the true distribution satisfies the constraint in (2.7) and the bound is exact.

ii) For distributions with small entropy, (2.10) leads to a tighter bound, which is exact for the extreme cases of minimum entropy $\mathcal{H}[\mathrm{P}] = 0$. Note that when $\mathcal{H}[\mathrm{P}] = 0$, all but one of the p_k is nonzero and a single value of k leads to a value of $V(d_k)+\epsilon\log(p_k)$ with $\epsilon > 0$ in (2.10) that is not $-\infty$.

Even though the DRO-based approach in (2.7)–(2.8) can often lead to a tighter bound than (2.9)–(2.10), an expected value $\sum_{k=1}^{K} V(u,d_k)\bar{p}_k$ still appears in (2.7)–(2.8) and therefore this bound is only helpful in simplifying computations if the optimal distribution $\bar{p}_1,\dots,\bar{p}_k$ has some particular structure that makes the computation of $\sum_{k=1}^{K} V(u,d_k)\bar{p}_k$ easier than the original computation $\mathbb{E}_{\mathrm{P}}[V(D)] = \sum_{k=1}^{K} V(u,d_k)p_k$.

Remark 2.1.7 *We focused this section on a discrete random variable D to avoid the technicalities that would arise from optimizations over general probability measures in* (2.7) *and* (2.10)*, but all the key observations made in this section remain unchanged for a continuous random variable D.* □

2.2 Stochastic Programming

We define the following *stochastic programming problem* with a single scalar constraint, but the approach proposed can easily be extended to multiple constraints: Let D be a random vector taking values in $\mathcal{D} \subset \mathbb{R}^M$. Given measurable functions $V : \mathcal{U}\times\mathcal{D} \mapsto \mathbb{R}$ and $G : \mathcal{U} \times \mathcal{D} \mapsto \mathbb{R}$, with $\mathcal{U} \subset \mathbb{R}^N$ we want to solve

$$V^* \coloneqq \inf_{u\in\mathcal{U}} \Big\{ \mathbb{E}[V(u,D)] : \mathbb{E}[G(u,D)] \leqslant 0 \Big\}. \tag{2.11}$$

The following results provides bounds on V^*, based on the bounds from Theorem 2.1.1.

Theorem 2.2.1 (Bounds to Stochastic Programming) *Consider three E-distributive groups $(\mathcal{P}_V, \oplus_V)$, $(\mathcal{P}_G, \oplus_G)$, $(\mathcal{P}, \oplus)$; functions $\alpha_V : \mathcal{D} \mapsto \mathcal{P}_V$, $\alpha_G : \mathcal{D} \mapsto \mathcal{P}_G$, $\alpha : \mathcal{D} \mapsto \mathcal{P}$*

and define

$$J_V(u,d) \coloneqq V(u,d) \oplus_V \alpha_V(d) \oplus_V \mathbb{E}[\neg\alpha_V(D)]$$

$$J_G(u,d) \coloneqq G(u,d) \oplus_G \alpha_G(d) \oplus_G \mathbb{E}[\neg\alpha_G(D)]$$

$$J(u,d,\lambda) \coloneqq (V(u,d) + \lambda G(u,d)) \oplus \alpha(d) \oplus \mathbb{E}[\neg\alpha(D)],$$

$\forall u \in \mathcal{U}, d \in \mathcal{D}, \lambda \geqslant 0$. *If* $\mathbb{E}[\neg\alpha_V(D)]$, $\mathbb{E}[\neg\alpha_G(D)]$, $\mathbb{E}[\neg\alpha(D)]$ *are finite, then* $V^\triangledown \leqslant V^* \leqslant V^\vartriangle$ *and* $V^* \leqslant V^\ddagger$*, with*

$$V^\triangledown \coloneqq \inf_{u\in\mathcal{U}} \Big\{ \big(\operatorname{ess\,inf} J_V(u,D)\big) : \operatorname{ess\,inf} J_G(u,D) \leqslant 0 \Big\} \tag{2.12}$$

$$V^\vartriangle \coloneqq \inf_{u\in\mathcal{U}} \Big\{ \big(\operatorname{ess\,sup} J_V(u,D)\big) : \operatorname{ess\,sup} J_G(u,D) \leqslant 0 \Big\} \tag{2.13}$$

$$V^\ddagger \coloneqq \inf_{u\in\mathcal{U}} \sup_{\lambda\geqslant 0} \operatorname{ess\,sup} J(u,D,\lambda). \tag{2.14}$$

Furthermore, if the infimum in the definition of $V^\vartriangle$ *is achieved at some* $u = u^\vartriangle$ *that is feasible for* (2.13)*, then* $u^\vartriangle$ *is also feasible for* (2.11)*. Additionally, if the infimum in the definition of* $V^\ddagger$ *is finite and achieved at some* $u = u^\ddagger$ *then* $u^\ddagger$ *is also feasible for* (2.11)*.* □

Theorem 2.2.1 guarantees that a solution $u^\vartriangle$ to the optimization (2.13) is feasible for the original stochastic program in (2.11) and provides performances guarantees for $u^\vartriangle$, in the sense that the expected value $\mathbb{E}[V(u^\vartriangle, D)]$ obtained using $u^\vartriangle$ will be away from the optimal V^* by no more than $V^\vartriangle - V^\triangledown$, which can be computed by solving the optimizations (2.12)–(2.13). Similarly, a solution $u^\ddagger$ to the optimization (2.14) is also guaranteed to be feasible and the expected value $\mathbb{E}[V(u^\ddagger, D)]$ obtained using $u^\ddagger$ will be away from the optimal V^* by no more than $V^\ddagger - V^\triangledown$, which can be computed by solving the optimizations (2.12), (2.14).

It is important to note that $\mathbb{E}[\neg\alpha_V(D)]$ and $\mathbb{E}[\neg\alpha(D)]$ are constants that do not depend on either u nor d, and therefore their values do not affect the optimizations in

(2.12)–(2.14). This means that, if one is not able to determine analytically $\mathbb{E}[\neg\alpha_V(D)]$ or $\mathbb{E}[\neg\alpha(D)]$, any errors in estimating these quantities will not introduce errors in determining $u^\triangle$ or $u^\ddagger$. This is specially relevant in large scale problems where obtaining accurate numerical estimates of $\mathbb{E}[\neg\alpha_V(D)]$ and $\mathbb{E}[\neg\alpha(D)]$ might be challenging.

As was the case for Theorem 2.1.1, the tightness of the bounds in Theorem (2.2.1) depends strongly on the choice of the groups, the functions $\alpha_\cdot$, the cost function, and the underlying random variable. Nevertheless, as we will see in the next section, the value of u that minimizes (2.11) and the value of u that minimizes (2.13) or (2.14) are often very close.

Proof of Theorem 2.2.1. In view of from Theorem 2.1.1, we have that $\text{ess inf } \ J_G(u, D) \leqslant \mathbb{E}[G(u, D)] \leqslant \text{ess sup } \ J_G(u, D)$, which guarantees that if $u = u^\triangle$ is feasible for (2.13), then $u^\triangle$ is also feasible for (2.11). Moreover,

$$\begin{aligned} &\inf_{u\in\mathcal{U}} \Big\{ \mathbb{E}[V(u, D)] : \text{ess inf } \ J_G(u, D) \leqslant 0 \Big\} \\ &\leqslant \inf_{u\in\mathcal{U}} \Big\{ \mathbb{E}[V(u, D)] : \mathbb{E}[G(u, D)] \leqslant 0 \Big\} \\ &\leqslant \inf_{u\in\mathcal{U}} \Big\{ \mathbb{E}[V(u, D)] : \text{ess sup } \ J_G(u, D) \leqslant 0 \Big\}. \end{aligned}$$

From Theorem 2.1.1, we can also conclude that $\text{ess inf } \ J_V(u, D) \leqslant \mathbb{E}[V(u, D)] \leqslant \text{ess sup } \ J_V(u, D)$, from which it follows that $V^\triangledown \leqslant V^* \leqslant V^\triangle$.

To establish that $V^\ddagger$ is also an upper bound on V^*, assume by contradiction that $V^\ddagger < V^*$, which means that there exists some $u \in \mathcal{U}$ such that $\text{ess sup}\, J(u, D, \lambda) < V^*$, $\forall \lambda \geqslant 0$. In view of Theorem 2.1.1, this would mean that $\mathbb{E}[V(u, D) + \lambda G(u, D)] < V^*$, $\forall \lambda \geqslant 0$, which is only possible if $\mathbb{E}[G(u, D)] \leqslant 0$ and consequently $\mathbb{E}[V(u, D)] < V^*$. The existence of such an u violates (2.11).

Finally note that if the infimum in the definition of $V^\ddagger$ is finite and achieved at some $u = u^\ddagger$, then we must have $\text{ess sup}\, J(u, D, \lambda) \leqslant V^* < \infty$, $\forall \lambda \geqslant 0$. Reasoning as in the paragraph above, this allow us to conclude that $\mathbb{E}[G(u^{\ddagger}, D)] \leqslant 0$ and therefore $u^{\ddagger}$ is feasible. ■

2.2.1 Combination with Monte Carlo methods

Any point u_{feasible} that is feasible for the optimization (2.11) can be used to construct an upper bound by using Monte Carlo averaging to compute

$$V^* \leqslant \mathbb{E}[V(u_{\text{feasible}}, D)] \approx \frac{1}{K} \sum_{k=1}^{K} V(u_{\text{feasible}}, d_k), \tag{2.15}$$

where the d_k are independent samples of D. Moreover, it is possible to control the error introduced by the Monte Carlo averaging by using a sufficiently large number of samples K. Essentially, to have an error smaller than δ with high probability we need

$$K \geqslant c \operatorname{Var}[V(u_{\text{feasible}}, D)]/\delta^2, \tag{2.16}$$

where the constant c is typically small and depends on the desired confidence for the bound [18].

Any point $u^{\vartriangle}$ that achieves $V^{\vartriangle}$ and in feasible for (2.13) is also feasible for (2.11) and can be used in (2.15) to construct an upper bound that is typically tighter than $V^{\vartriangle}$, provided that K is sufficiently large; the same reasoning is true for $u^{\ddagger}$ and $V^{\ddagger}$. In fact, one can use Theorem 2.1.1 to compute other feasible points that may provide tighter upper bounds. For example, an alternative feasible point can be obtained by minimizing a lower bound on the criterion constrained by an upper bound on the constraints, which leads to

$$V^{\perp} \coloneqq \inf_{u \in \mathcal{U}} \Big\{ \big(\operatorname{ess\,inf} J_V(u, D) \big) : \big(\operatorname{ess\,sup} J_G(u, D) \leqslant 0 \big) \Big\}. \tag{2.17}$$

Unlike $V^{\vartriangle}$ and $V^{\triangledown}$ in Theorem 2.2.1, $V^{\perp}$ neither provides an upper nor a lower bound on V^*. However, any point that achieves the infimum and is feasible for (2.17) is also feasible for (2.11) and therefore can be used to construct the upper bound in (2.15). An alternative method to combine the results in Theorem 2.1.1 with Monte Carlos methods is obtained by replacing the optimization in (2.14) by

$$\inf_{u \in \mathcal{U}} \operatorname{ess\,sup} J(u, D, \lambda)$$

for some fixed $\lambda \geqslant 0$. Rather than taking the supremum over $\lambda \geqslant 0$ that appears in (2.14), one could simply adjust λ and/or artificially tightening the constraint until a Monte Carlo estimate for $\mathbb{E}[G(u, D)]$ guarantees that the constraint is satisfied with a sufficiently large confidence.

Remark 2.2.2 (Contrast with Sample Average Approximation) *It is important to emphasize the difference between using Monte Carlo averaging to estimate the value of the expected value for a given value of* $u_{\text{feasible}} \in \mathcal{U}$*, as in* (2.15)*, and optimizing a Monte Carlo approximation of the criterion, as in*

$$\min_{u \in \mathcal{U}} \frac{1}{K} \sum_{k=1}^{K} V(u, d_k), \tag{2.18}$$

which is typically referred to as the Sample Average Approximation (SAA). We can see in (2.16) *that the number of samples required to achieve a desired error* $\delta > 0$ *depends mostly on the variance of* $V(u, D)$ *at the point* u_{feasible}*. However, the sample complexity required to obtain the same error in* (2.18) *is typically much larger as the numerator of* (2.16) *would be determined by the Vapnik-Chervonenkis (VC) dimension of the family of functions* $u \mapsto \mathbb{E}[V(u, D)]$ *[95, 96].* □

2.2.2 Numerically computing the bounds

We show next that under appropriate regularity assumptions, the essential infima and suprema in (2.12) and (2.13) are achieved at minima and maxima, respectively, and can be computed using numerical solvers. To formalize this observation we recall that a function $f : \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subset \mathbb{R}^n$ is said to have *compact sublevel sets* if its sublevel sets $\{x \in \mathcal{X} : f(x) \leqslant \lambda\}$ are compact for every finite $\lambda \in \mathbb{R}$ and it is said to have *compact suplevel sets* if $-f$ has compact sublevel sets. For the remainder of this section we assume that $\mathcal{U} \subset \mathbb{R}^N$ and that $\mathcal{D} \subset \mathbb{R}^M$ is the support of the random variable D, i.e., the smallest subset of $\mathbb{R}^M$ for which $\mathrm{P}(d \in \mathcal{D}) = 1$.

Theorem 2.2.3 *Assume that the functions $J_V, J_G : \mathcal{U} \times \mathcal{D} \to \mathbb{R}$ are continuous and that $\mathcal{U}$ is compact. If J_V and J_G have compact sublevel sets and* (2.12) *is feasible, then $V^{\triangledown}$ can be obtained by solving*

$$V^{\triangledown} = \min_{u \in \mathcal{U}, d, \tilde{d} \in \mathcal{D}} \Big\{ J_V(u, d) : J_G(u, \tilde{d}) \leqslant 0 \Big\}. \tag{2.19}$$

If $J_V(u, d)$ and $J_G(u, d)$ have compact suplevel sets and (2.13) *is feasible, then $V^{\vartriangle} = \lim_{\mu \to \infty} V^{\vartriangle}_{\mu}$, with*

$$V^{\vartriangle}_{\mu} = \min_{u \in \mathcal{U}} \max_{d, \tilde{d} \in \mathcal{D}} J_V(u, d) + \mu \Big(\max \big\{ 0, J_G(u, \tilde{d}) \big\} \Big)^2. \tag{2.20}$$

Whenever the minimum and maxima in (2.20) *are achieved for values $u^{\vartriangle} \in \mathcal{U}$ and $d^{\vartriangle}, \tilde{d}^{\vartriangle} \in \mathcal{D}$, respectively, for which $J_G(u^{\vartriangle}, \tilde{d}^{\vartriangle}) \leqslant 0$, then $u^{\vartriangle}$ is feasible and $V^{\vartriangle}_{\mu}$ is an upper bound for V^*.* □

The minimization in (2.19) is a regular constrained optimization and can be solved using commercial products like Knitro [97] or open-source solvers like IPOPT [98] and `TensCalc` [99]. For the sequence of minmax problem in (2.20), different algorithms are applicable depending on the convexity assumptions (convex-concave, nonconvex-concave, convex-nonconcave, nonconvex-nonconcave). These include methods based on robust counterpart [26, 58–60], cutting-set [61], and variations of gradient descent-ascent methods such as [65,66,70,73,75,100–105] among many others. For the examples in Section 2.3, we used `TensCalc`, which is based on a variation of interior point methods for gradient descent ascent.

Regardless of the solver used, for nonconvex problem like the ones typically arising in (2.19)–(2.20), it is generally hard to be certain that a local minimum or a local minmax [32] found by a numerical solver is actually a global optimum. An approach that can be used to obviate this problem is to replace the nonconvex optimization that arises from our bounds by pessimistic or optimistic convex relaxation, depending on whether we are interested in an upper or lower bound on the expected value, respectively. An alternative approach relies on analyzing the consequences of a solver getting stuck at a local optima and responding to the specific problems encountered: Theorem 2.2.3 essentially proposes to replace the stochastic optimization in (2.11) by the sequence of deterministic robust optimizations $V_\mu^\triangle$. A numerical solver for (2.20) can typically be "fooled" in three ways:

i) The solver could converge to a value $d^\triangle$ for d that is a local but not a global extremum to the inner maximization. This would mean that the value $V_\mu^\triangle$ returned by the solver is actually not an upper bound on $\mathbb{E}[V(u^\triangle, D)]$. If it is important to obtain a high-confidence bound for this expected value and the inner maximization is not concave (or known to only have a unique local/global maximum), then one can use a Monte Carlo method to get an accurate estimate for $\mathbb{E}[V(u^\triangle, D)]$, which is typically computationally much easier than solving (2.11), as discussed in Section 2.2.1.

ii) The solver may converge to a value $u^\triangle$ for u for which $J_G(u^\triangle, \tilde{d}^\triangle) \leqslant 0$ holds for a local maximum $\tilde{d}^\triangle$ that is not global and the expected value $\mathbb{E}[G(u^\triangle, D)]$ is actually positive. Again here, once the optimization finishes, we can use a Monte Carlo method to obtain an accurate estimate for $\mathbb{E}[G(u^\triangle, D)]$ and reject the solution $u^\triangle$ if the constraint is violated. Hopefully, different initialization for the solver would resolve this, but one could also tighten the constraint by asking $\max_{\tilde{d}\in\mathcal{D}} J_G(u, \tilde{d})$ to actually be negative.

iii) Finally the solver, may return a value $u^\triangle$ for u that satisfies the constraint but is a local (rather than a global) extremum of the outer minimization. In this case, it may be possible to get a better solution, but the solver was unable to find it. In practice, for nonconvex problems there is little protection against this, rather than trying a different initialization for the solver.

It should be noted that any approach based on constructing (non-exact) convex relaxations to (2.20) will have very similar issues: pessimistic relaxations may overlook better solutions (as in iii), whereas optimistic relaxations may accept solutions that violate constraints (as in ii).

Lemma 2.2.4 (Equivalent compact subset) *Consider a continuous function $J : \mathcal{U} \times \mathcal{D} \to \mathbb{R}$ with $\mathcal{U}$ compact. If J has compact sublevel sets, there exists a compact set $\mathcal{D}^\dagger \subset \mathcal{D}$ such that*

$$J^{\inf}(u) \coloneqq \operatorname*{ess\,inf}_{d \in \mathcal{D}} J(u,d) = \min_{d \in \mathcal{D}^\dagger} J(u,d), \quad \forall u \in \mathcal{U}$$

and the function $J^{\inf}$ is continuous. Similarly, if J has compact suplevel sets, there exists a compact set $\mathcal{D}^\dagger \subset \mathcal{D}$ such that

$$J^{\sup}(u) \coloneqq \operatorname*{ess\,sup}_{d \in \mathcal{D}} J(u,d) = \max_{d \in \mathcal{D}^\dagger} J(u,d), \quad \forall u \in \mathcal{U}$$

and the function $J^{\sup}$ is continuous.

Proof of Lemma 2.2.4. First note that because $\mathcal{D}$ is the support of the random variable D and J is continuous, the essential infimum of $J(u,D)$ is equal to the usual infimum of $J(u,d)$ over $d \in \mathcal{D}$. The same is true for the supremum,

We prove the result only for the minimization, as the proof for the maximization is analogous. Take an arbitrary point $d^\dagger \in \mathcal{D}$ and define

$$\lambda^\dagger \coloneqq \max_{u \in \mathcal{U}} J(u, d^\dagger), \qquad \mathcal{S}^\dagger \coloneqq \{(u,d) \in \mathcal{U} \times \mathcal{D} : f(u,d) \leqslant \lambda^\dagger\}.$$

The constant $\lambda^\dagger$ is finite because J is continuous and $\mathcal{U} \times \{d^\dagger\}$ is a compact set, and the set $\mathcal{S}^\dagger$ is compact because J has compact sublevel sets. The desired set $\mathcal{D}^\dagger$ is then given by the closure of

$$\mathcal{D}_o \coloneqq \bigcup_{u \in \mathcal{U}} \Big\{ d \in \mathcal{D} : (u, d) \in \mathcal{S} \Big\}.$$

Note that $\mathcal{D}_o$ is bounded because $\mathcal{S}$ is bounded and therefore its closure $\mathcal{D}^\dagger$ is compact. To show that the infimum of $J(u, d)$ over $\mathcal{D}$ is achieved at some point in $\mathcal{D}^\dagger$, assume by contradiction that there exists some $d^* \notin \mathcal{D}^\dagger$ such that $J(u, d^*) < J(u, d)$, $\forall d \in \mathcal{D}^\dagger$. Since $d^\dagger \in \mathcal{D}^\dagger$, we conclude that $J(u, d^*) < J(u, d^\dagger) \leqslant \lambda^\dagger$. This establishes a contradiction, because it would mean that $(u, d^*) \in \mathcal{S}^\dagger$ and therefore $d^* \in \mathcal{D}_o \subset \mathcal{D}^\dagger$. Continuity of $J^{\inf}$ then follows from Berge's Maximum Theorem [106, Chapter E.3]. ■

Proof of Theorem 2.2.3. In view of Lemma 2.2.4, all the essential infima and suprema in (2.12)–(2.13) are achieved at some point inside a compact subset $\mathcal{D}^\dagger$ of $\mathcal{D}$ and

$$V^\triangledown = \inf_{u \in \mathcal{U}} \left\{ J_V^{\inf}(u) : J_G^{\inf}(u) \leqslant 0 \right\} \tag{2.21}$$

for the continuous functions

$$J_V^{\inf}(u) \coloneqq \min_{d \in \mathcal{D}} J_V(u, d), \quad J_G^{\inf}(u) \coloneqq \min_{d \in \mathcal{D}} J_G(u, d). \tag{2.22}$$

Since $J_G^{\inf}(u)$ is continuous and $\mathcal{U}$ is compact, the feasible set $\{u \in \mathcal{U} : J_G^{\inf}(u) \leqslant 0\}$ is compact and nonempty by assumption. Weierstrass Theorem [107, Proposition A.8] then allow us to conclude that the inf is actually achieved at some point $u^\triangledown \in \mathcal{U}$ of the feasible set. Denoting by $d_V^\triangledown$ and $d_G^\triangledown$ points in $\mathcal{D}$ at which the minima in (2.22) are achieved for $u = u^\triangledown$, we conclude that

$$V^\triangledown = J_V(u^\triangledown, d_V^\triangledown), \quad J_G(u^\triangledown, d_G^\triangledown) \leqslant 0,$$

which shows that the right-hand side of (2.19) cannot be larger than $V^\triangledown$. By contradiction, assume that it is actually strictly smaller than $V^\triangledown$. This would mean there exist $u \in \mathcal{U}$ and $d, \tilde{d} \in \mathcal{D}$ such that

$$J_v(u,d) < V^\triangledown, \quad J_G(u,\tilde{d}) \leqslant 0.$$

The right-hand side inequality shows that $J_G^{\inf}(u) \leqslant 0$ and therefore such u is feasible for (2.21) and the left-hand side inequality shows that $J_V^{\inf}(u) < V^\triangledown$, which contradicts the fact that the infimum in (2.21) is equal to $V^\triangledown$.

Again using Lemma 2.2.4, we conclude that

$$V^\vartriangle = \inf_{u \in \mathcal{U}} \left\{ J_V^{\sup}(u) : J_G^{\sup}(u) \leqslant 0 \right\} \tag{2.23}$$

for the continuous functions

$$J_V^{\sup}(u) \coloneqq \max_{d \in \mathcal{D}} J_V(u,d), \quad J_G^{\sup}(u) \coloneqq \max_{d \in \mathcal{D}} J_G(u,d). \tag{2.24}$$

In view of [107, Proposition 4.2.1], $\lim_{\mu \to \infty} \bar{V}_\mu^\vartriangle = V^\vartriangle$, with

$$\bar{V}_\mu^\vartriangle \coloneqq \min_{u \in \mathcal{U}} J_V^{\sup}(u) + \mu \Big(\max \left\{0, J_G^{\sup}(u)\right\} \Big)^2.$$

The result then follows by noting that

$$\max \left\{0, J_G^{\sup}(u)\right\} = \max_{\tilde{d} \in \mathcal{D}} \max \left\{0, J_G(u,\tilde{d})\right\}.$$

and therefore $\bar{V}_\mu^\vartriangle = V_\mu^\vartriangle$ for positive μ. ■

2.3 Selected Applications

2.3.1 Stochastic control

Consider the dynamical system

$$x_{t+1} = f\big(x_t, \theta, u_t, d_t\big) \tag{2.25a}$$

$$y_t = h\big(x_t\big) + n_t, \tag{2.25b}$$

where x_t denotes the state of the system at time t, u_t the controlled input, d_t a random disturbance input, y_t the measured output, n_t measurement noise, and θ a random vector of parameters.

Our goal to select control inputs $u_0, \dots, u_{T-1}$ to minimize a finite-horizon criterion of the form

$$\mathbb{E}\big[W(x_1, \dots, x_T, u_0, \dots, u_{T-1})\big], \tag{2.26}$$

subject to a constraint of the form

$$\mathbb{E}\big[U(x_1, \dots, x_T, u_0, \dots, u_{T-1})\big] \leqslant 0. \tag{2.27}$$

We consider two versions of this problem: First a *state-feedback* scenario in which the initial state x_0 is known and the expectation (2.26) is with regard to the random parameters θ and the disturbances $d_0, \dots, d_{T-1}$. We then consider an *output-feedback* scenario in which the initial state is not known, but one has available past measurements $y_{-K}, \dots, y_0$. In this case, the expectation in (2.26) is conditioned to these past measurements and it regards the measurement noise $n_{-K}, \dots, n_0$, the initial state x_{-K}, and the past disturbances $d_{-K}, \dots, d_{-1}$.

State Feedback The state-feedback control problem can be viewed as an instance of (2.11), with the following associations

$$\begin{aligned}
u &\coloneqq \big(u_0, \dots, u_{T-1}\big), \\
D &\coloneqq \big(\theta, d_0, \dots, d_{T-1}\big), \\
V(u, D) &\coloneqq W(x_1, \dots, x_T, u_0, \dots, u_{T-1}), \\
G(u, D) &\coloneqq U(x_1, \dots, x_T, u_0, \dots, u_{T-1}),
\end{aligned}$$

with the understanding that the states $x_1, \dots, x_T$ that appear in the definitions of $V(u,D)$ and $G(u,D)$ are obtained along solutions to (2.25a) for the control input in u and the parameters and input disturbances in D.

Assuming that the disturbances d_t are independent and identically distributed with *pdf* $p_d(\cdot)$ and differential entropy $\mathcal{H}_d$, and that the parameter θ has *pdf* $p_\theta(\cdot)$ and differential entropy $\mathcal{H}_\theta$, we have that

$$\mathcal{H}_D = \mathcal{H}_\theta + T\mathcal{H}_d,\ \log p_D(\theta, d) = \log p_\theta(\theta) + \sum_{t=0}^{T-1} \log p_d(d_t),$$

and the optimization in (2.13) with additive upper bounds for $\oplus_G$ and $\oplus_V$ takes the form

$$\begin{aligned} V^\triangle &= \min_{u \in \mathcal{U}} \left\{ X(u) : U(\bar{x}_1, \dots, \bar{x}_T, u_0, \dots, u_{T-1}) + \bar{\epsilon}\mathcal{H}_D + +\bar{\epsilon}\log p_D(\bar{\theta}, \bar{d}) \leqslant 0,\ \forall \bar{\theta}, \bar{d} \right\} \\ X(u) &\coloneqq \max_{\theta \in \Theta, d \in \mathcal{D}} W(x_1, \dots, x_T, u_0, \dots, u_{T-1}) + \epsilon\mathcal{H}_D + \epsilon \log p_D(\theta, d), \end{aligned}$$

where $\mathcal{U}$ denotes the set of admissible controls; Θ and $\mathcal{D}$ the supports of the distributions for the random parameter and disturbance, respectively; $\bar{x}_1, \dots, \bar{x}_T$ the solution to (2.25a) for the control $u \coloneqq (u_0, \dots, u_{T-1})$, parameter $\bar{\theta}$ and disturbance $\bar{d} \coloneqq (\bar{d}_0, \dots, \bar{d}_{T-1})$; $x_1, \dots, x_T$ the solution to (2.25a) for the same control $u \coloneqq (u_0, \dots, u_{T-1})$, but parameter θ and disturbance $d \coloneqq (d_0, \dots, d_{T-1})$; and ϵ, $\bar{\epsilon}$ the scalar parameters associated with additive upper bounds used for $\oplus_G$ and $\oplus_V$, respectively. An equivalent formulation of the optimization in (2.12) gives $V^\triangledown$.

Output Feedback The output-feedback problem can also be viewed as an instance of (2.11), but now with the following associations

$$\begin{aligned} &u \coloneqq \big(u_0, \dots, u_{T-1}\big), \\ &D \coloneqq \big(\theta, x_{-K}, d_{-K}, \dots, d_{T-1}\big), \\ &V(u, D) \coloneqq W(x_1, \dots, x_T, u_0, \dots, u_{T-1}), \end{aligned}$$

$$G(u,D) \coloneqq U(x_1,\dots,x_T,u_0,\dots,u_{T-1}),$$

with the understanding that the states $x_1,\dots,x_T$ that appear in the definition of $V(u,D)$ and $G(u,D)$ are obtained along solutions to (2.25a) for the control input in u and the parameters, initial state, and input disturbances in D. In addition, the expectation in (2.11) is now a conditional expectation, given measurements $Y=(y_{-K},\dots,y_0)$ defined by (2.25b).

In this case, the optimization in (2.13) with additive upper bounds for $\oplus_V$ and $\oplus_G$ takes the form

$$V^{\triangle} = \min_{u\in\mathcal{U}} \Big\{ X(u) : U(\bar{x}_1,\dots,\bar{x}_T,u_0,\dots,u_{T-1}) + \\ \bar{\epsilon}\log p_{D|Y}(\bar{\theta},\bar{d},\bar{x}_k) + \bar{\epsilon}\mathcal{H}_{D|Y}(y_{-K},\dots,y_0) \leqslant 0, \forall \bar{\theta},\bar{d},\bar{x}_k \Big\}$$

$$X(u) \coloneqq \max_{\theta\in\Theta, d\in\mathcal{D}, x_{-K}\in\mathcal{X}_{-K}} W(x_1,\dots,x_T,u_0,\dots,u_{T-1}) + \\ \epsilon\log p_{D|Y}(\theta,d,x_{-K}) + \epsilon\mathcal{H}_{D|Y}(y_{-K},\dots,y_0), \quad (2.28)$$

where we use the version of the bounds for conditional expectation mentioned in Remark 2.1.2. The conditional *pdf* that appears in (2.28) can be computed using the following result.

Lemma 2.3.1 (Conditional *pdf* of a dynamical system) *In addition to the assumptions made for the state feedback case, also assume that the observation noises n_t are independent and identically distributed with pdf $p_n(\cdot)$ and that the initial state x_{-K} has pdf $p_{x_{-K}}(\cdot)$. If $p_Y(y_{-K},\dots,y_0)\neq 0$, the conditional probability density function $p_{D|Y}(\cdot)$ is given by*

$$\frac{\prod_{t=-K}^{0} p_n\Big(y_t - h\big(x_t\big)\Big) \prod_{t=-K}^{T-1} p_d(d_t) p_{x_{-K}}(x_{-K})\, p_\theta(\theta)}{p_Y(y_{-K},\dots,y_0)}$$

with the understanding that x_t is obtained along the solutions to (2.25a). □

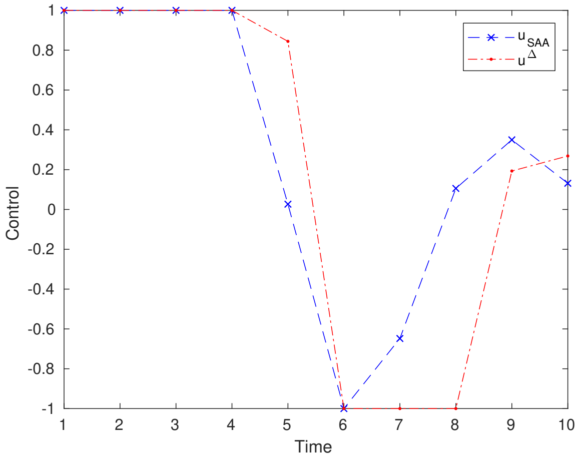


Figure 2.1: Linear system with unknown dynamics, comparison of the controls $u^{\triangle}$ obtained from Theorem 2.2.1 and u_{SAA} obtained using Sample Average Approximation. Using Monte Carlo integration, we obtain that $\mathbb{E}[V(u^{\triangle}, D)] = 1.79 \times 10^3$ and $\mathbb{E}[V(u_{SAA}, D)] = 1.62 \times 10^3$.

Proof of Lemma 2.3.1. Using the independence of n_t, one deduces that the observations y_t are conditionally independent:

$$p_{Y|D}(y_{-K}, \dots, y_0 \mid x_{-K}, \dots, x_0) = \prod_{t=-K}^{0} p_{Y_t|D}(y_t \mid x_t).$$

As the noise n_t is additive in (2.25b), a change of variable gives $p_{Y_t|D}(y_t \mid x_t) = p_n\Big(y_t - h\big(x_t\big)\Big)$. Using Bayes' theorem and the independence of d_t, θ, and x_{-K} finishes the proof. ■

The differential entropy $\mathcal{H}_{D|Y}(y_{-K}, \dots, y_0)$ that appears in (2.28) is typically difficult to compute (or even to estimate, e.g., through Monte Carlo integration); especially for a long sequence of *past* measurements $y_{-K}, \dots, y_0$. However, this entropy is not affected by the optimization variable $u = (u_0, \dots, u_{T-1})$, which only includes *future* controls. This means that we can determine the optimal value for u in (2.28) without actually computing $\mathcal{H}_{D|Y}(y_{-K}, \dots, y_0)$.

Example 2.3.2 (Linear system with unknown dynamics) *Consider a linear sys-*

tem, i.e., a system with dynamics

$$x_{t+1} = A\,x_t + B\,u_t + d_t$$

$$y_t = C\,x_t + n_t$$

with d_t *and* n_t *independent zero mean standard Gaussian processes. The system is time-invariant,* C *is an identity matrix, but the matrices* A *and* B *are unknown stochastic parameters of the form*

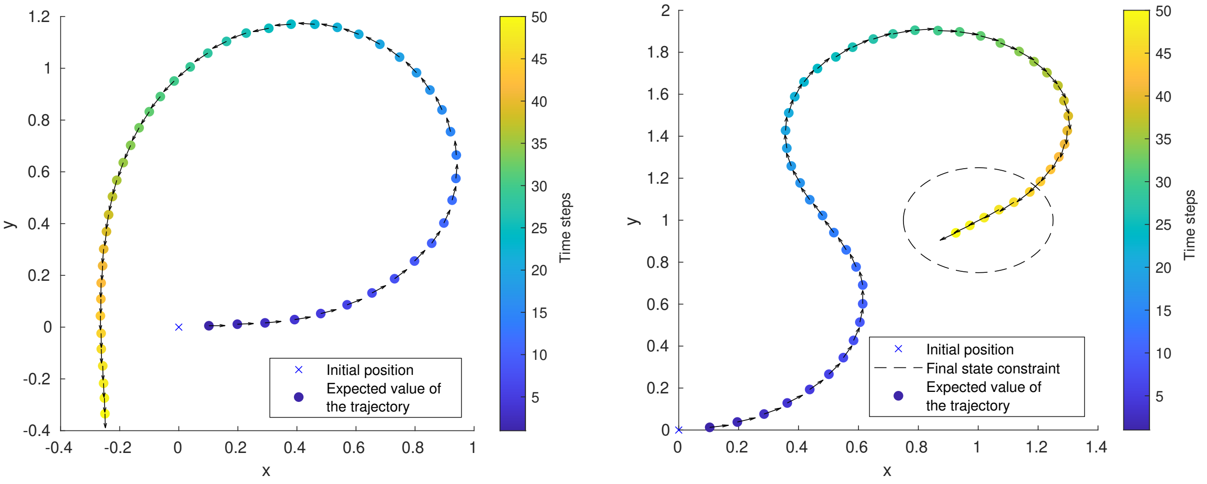
$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \end{bmatrix} \; B = \begin{bmatrix} 0 \\ 0 \\ B_{31} \end{bmatrix},$$

where A_{11}*,* A_{12}*,* A_{22}*,* A_{23}*,* A_{33}*,* B_{31} *are independent Gaussian random variables with mean* 1 *and standard deviation* 0.25*. We chose a quadratic cost*

$$W(u_0, \dots u_{T-1}, x_0 \dots x_T) = \sum_{t=0}^{T-1} 0.5\|u_t\|_2^2 + 0.5\|x_t\|_2^2 + 0.5\|x_T\|_2^2$$

with a future horizon $T = 10$ *and constraints on the control that* $\|u\|_\infty \leqslant 1$*. We suppose access to past measurements* $y_{-K}, \dots, y_0$ *with* $K = 20$*.*

The value of the upper bound $V^\vartriangle$ *is* 5.04×10^5 *and the value of the lower bound* $V^\triangledown$ *is* 28*. We compare our results with an approximate solution obtained using Sample Average Approximation (SAA) (i.e., minimizing an empirical mean of the cost). Solving the upper bound and lower bound optimizations (Theorem 2.2.1) takes about* 0.1 *seconds, while solving the Sample Average Approximation takes about 5 minutes. In Figure 2.1 one can see that the controls match each other fairly closely until* $t = 6$*, when they start to slightly diverge. We also use Monte Carlo integration, as discussed in Section 2.2.1, to estimate the expected value of the cost for the two controls, obtaining that they differ by about* 10%*.*



(a) Expected value of the trajectory without constraints on the final state; the expected value of the cost for this control is $\mathbb{E}(V(u^{\Delta}, D)) = 189$.

(b) Expected value of the trajectory with constraints on the final state; the expected value of the cost for this control is $\mathbb{E}(V(u^{\ddagger}, D)) = 250$.

Figure 2.2: Expected value of the trajectory of the Dubins vehicle given two different controls estimated using Monte Carlo integration. Without constraints (a), the control brings the expected value of the trajectory back to near the origin. With the inclusion of constraints (b), the control drives the expected value of the final state towards the correct region.

Example 2.3.3 (Dubins vehicle) *Consider a discrete time Dubins vehicle [108, 109] with dynamics*

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \omega_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \omega_t \end{bmatrix} + T_s \begin{bmatrix} v\,\cos(\omega_t) \\ v\,\sin(\omega_t) \\ u_t \end{bmatrix} + \frac{T_s^2}{2} \begin{bmatrix} -v\,\sin(\omega_t)\,u_t \\ v\,\cos(\omega_t)\,u_t \\ 0 \end{bmatrix} + d_t$$

where $T_s = 0.1$ is the sampling period, $v = 1$ is a constant forward speed. The initial state is known to be $[x_0, y_0, \omega_0]' = [0, 0, 0]$, and we want to optimize for a future horizon $T = 50$. The controls are constrained such that $\|u\|_\infty \leqslant \pi/2$. The disturbance $d_t = [d_t^{(x)}, d_t^{(y)}, d_t^{(\omega)}]'$ is such that $d_t^{(x)}, d_t^{(y)}$ are zero mean Gaussian random variables with variance T_s, and $d_t^{(\omega)}$ is a von Mises random variable, with probability density function $e^{\kappa \cos(x)}/(2\pi I_0(\kappa))$ with

$\kappa = 5/T_s$ and where $I_0(\kappa)$ is the modified Bessel function of order 0. The cost function is

$$W(u_0, \dots u_{T-1}, x_0 \dots x_T) = \sum_{t=0}^{T-1} 0.5\|u_t\|_2^2 + \sum_{t=0}^{T} 0.5\|x_t\|_2^2 + 0.5\|y_t\|_2^2$$

We present two cases, one with no constraints on the states and one with a constraint on the final state. For both of them, we use the additive bounds.

The first case, without constraints, takes about 1 second to solve, the value of the upper bound $V^{\triangle}$ is 1.25×10^5 the lower bound $V^{\triangledown}$ only provides the trivial value of 0. However, using a Monte Carlo integration we compute the expected value of the cost given the control and obtain 189. We use a Stochastic Gradient Descent to solve (2.26), which takes about 15 seconds, the optimal cost is 187 and the error between the solution obtained using the Stochastic Gradient Descent u_{SGD} and the solution obtained using the upper bound $u^{\triangle}$ is $\|u_{SGD} - u^{\triangle}\|_{\infty} = 0.029$, suggesting that $u^{\triangle}$ approximately finds the optimal solution to (2.26)

For the second case, we include the constraint

$$\mathbb{E}\left[\left\|\begin{bmatrix} x_T \\ y_T \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\|_2\right] \leqslant 0.25 \tag{2.29}$$

i.e., we want to find a control such that the expectation of the final value of the trajectories of (x, y) be in neighborhood around the point $(1, 1)$ (look at Figure 2.2b for a visualization of the constraints). As the problem now has stochastic constraint, we have to choose between using the upper bound $V^{\triangle}$ from (2.13) which requires u to satisfy the constraint

$$\max_d \left\|\begin{bmatrix} x_T \\ y_T \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\|_2 + \epsilon \log p_D(d) + \epsilon \mathcal{H}_D \leqslant 0.25, \tag{2.30}$$

or the upper bound $V^{\ddagger}$ from (2.14). Unfortunately the bound (2.30) of (2.29) is too conservative, and renders the problem infeasible. The upper bound $V^{\ddagger}$ does not suffer from this problem. It takes about 30 seconds to solve the optimization for which we *obtain a value for the upper bound* $V^{\ddagger}$ *of* 1.25×10^8 *and the lower bound* $V^{\triangledown}$ *provides only the trivial value of 0. However, using Monte Carlo integration, we obtain that the expected value of the cost is* 250.

2.3.2 Maximum Likelihood and Maximum a Posteriori with latent variables

Consider an observation x of a random vector X taking values in $\mathbb{R}^M$ whose distribution depends on an unknown parameter $\theta \in \mathbb{R}^P$ that one wants to estimate. The Maximum Likelihood Estimation (MLE) [9] of θ is a vector $\theta^* \in \mathbb{R}^P$ such that

$$\theta^* \in \arg\max_{\theta} p_X(x;\theta). \tag{2.31}$$

where the *pdf*of X is $p_X(x;\theta)$. The Maximum a Posteriori (MAP) is the analogous of the MLE in Bayesian estimation, *i.e.*, when one regards θ as a realization of a random variable Θ, called the prior, which has *pdf*$p_\Theta(\cdot)$. In this case, the MAP estimation of θ is a vector $\theta^* \in \mathbb{R}^P$ such that

$$\theta^* \in \arg\max_{\theta} p_{X|\Theta}(x \mid \theta)p_\Theta(\theta). \tag{2.32}$$

In many cases, constructing the model requires including latent variables that cannot be directly observed. This means that one does not know $p_X(x;\theta)$ but does know $p_{X|D}(x \mid d;\theta)p_D(d)$, where D is a "latent" random vector taking values in $\mathbb{R}^N$. In this case, the MLE θ^* is given by

$$\theta^* \in \arg\max_{\theta} p_X(x;\theta) = \int_{\mathcal{D}} p_{X|D}(x \mid d;\theta)p_D(d)\,\mathrm{d}d = \arg\max_{\theta} \mathbb{E}[p_{X|D}(x \mid D;\theta)]. \tag{2.33}$$

For the MAP, the analogous deduction leads to

$$\theta^* \in \arg\max_{\theta} \mathbb{E}\big[p_{X|D}(x \mid D;\theta)\big]p_\Theta(\theta). \tag{2.34}$$

Computing the expected values in (2.33) or in (2.34) is normally intractable. The standard approach is to use the Expectation Maximization (EM) algorithm [110]. An issue with EM, in addition to a rate of convergence that might be very slow, is that it requires computing in closed form the expected value $\mathbb{E}_{D|X;\tilde{\theta}}[\log p_{X,D}(X, D; \theta)]$, which is often not possible. In some cases, one can use Monte Carlo EM [110] to compute it, but with rates of convergence even slower.

The MLE optimization (2.33) can be viewed as an unconstrained form of (2.11), which using the multiplicative upper bound in (2.13) leads to

$$\theta^{\triangledown} \in \arg\max_{\theta}\min_{d} p_{X|D}(x\,|\,d;\theta)p_D(d)^{\epsilon}\,\mathcal{I}_D(\epsilon) \tag{2.35}$$

or equivalently,

$$\theta^{\triangledown} \in \arg\max_{\theta}\min_{d} \log\bigl(p_{X|D}(x\,|\,d;\theta)\,p_D(d)^{\epsilon}\,\mathcal{I}_D(\epsilon)\bigr), \tag{2.36}$$

which is numerically more stable. For the MAP, one would add $\log p_{\Theta}(\theta)$ to the right hand side of (2.36). The multiplicative bound is more amenable for the optimization than the additive as it allows to solve (2.35) in its logarithmic form (2.36).

Example 2.3.4 (Linear measurements with additive Gaussian noise) *Let $D \sim \mathcal{N}(0, \sigma_D)$, $N \sim \mathcal{N}(0, \sigma_N)$. Consider T observations of the random variable $X_t = \theta + D_t + N_t$ where θ is the parameter to be estimated. This problem has a closed form solution, which is the empirical average of x_t. Applied to this problem, equation (2.36) reduces to* [1]

$$\theta^{\triangledown} \in \arg\max_{\theta}\min_{d_{1:T}} \sum_{t=1}^{T} -\|x_t - d_t - \theta\|_2^2\sigma_N^{-1} - \epsilon\|d_t\|_2^2\sigma_D^{-1} - T\log(1-\epsilon) - T\log(2\pi\sigma_N).$$

If we take any ϵ such that $\epsilon < -\sigma_N^{-1}/\sigma_D^{-1}$, then the solution is $\frac{1}{T}\sum_{t=1}^{T} x_t$ which is the same as the exact solution.

[1] We refer the reader to Appendix 2.A.1 for the deduction of the penalizing term.

θ (actual value)	$\theta^{\triangledown}$	naive MAP	MC MMSE	MC MAP
$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1.95 \\ 2.37 \end{bmatrix}$	$\begin{bmatrix} 2.8 \\ 0.21 \end{bmatrix}$	$\begin{bmatrix} 1.82 \\ 1.62 \end{bmatrix}$	$\begin{bmatrix} 2.13 \\ 1.70 \end{bmatrix}$

Table 2.1: Comparison between the actual value of θ, of $\theta^{\triangledown}$ obtained from (2.37) and three other estimators.

Example 2.3.5 (Norm measurements with Gaussian disturbances and noise) *We have T observations of the random variable $X_i = \|\theta + D_i\|_2 + N_i$ where θ is the parameter to be estimated, $D \sim \mathcal{N}(0, \Sigma_D)$ and $N \sim \mathcal{N}(0, \sigma_N)$. We also have a prior distribution $\Theta \sim \mathcal{N}(\bar{\theta}, \Sigma_\Theta)$ on θ. Applied to this problem, equation (2.36) reduces to*

$$\begin{aligned}\theta^{\triangledown} \in \arg\max_{\theta} \min_{d_{1:T}} \sum_{t=1}^{T} - \Big\| x_t - \|\theta + d_t\|_2 \Big\|^2_{\sigma_N^{-1}} - \epsilon \|d_t\|^2_{\Sigma_D^{-1}} \\ - \Big\| \theta - \bar{\theta} \Big\|^2_{\Sigma_\Theta^{-1}} - 2T \log(1-\epsilon) - T \log(2\pi\sigma_N) \quad (2.37)\end{aligned}$$

where we use the notation $\|v\|^2_Q \coloneqq v'Qv$. We take the numerical values $T = 20$, $\Sigma_D = \left[\begin{smallmatrix} 2 & -1 \\ -1 & 1 \end{smallmatrix}\right]$, $\sigma_N = 1$ $\bar{\theta} = \left[\begin{smallmatrix} 1.8 \\ 1.8 \end{smallmatrix}\right]$, Σ_Θ the identity matrix.

The result of (2.37) *is shown in Table 2.1 where we compare it with three other estimators. The first one is what we call naive MAP, where one treats $D_1, \ldots, D_T$ not as a latent variable, but as a regular variable that one wants to estimate, i.e.,*

$$\arg\max_{\theta, d_{1:T}} \sum_{t=1}^{T} - \Big\| x_t - \|\theta + d_t\|_2 \Big\|^2_{\sigma_N^{-1}} - \|d_t\|^2_{\Sigma_D^{-1}} - \Big\| \theta - \bar{\theta} \Big\|^2_{\Sigma_\Theta^{-1}}.$$

The second and third are Monte Carlo methods, where we use a Markov Chain Monte Carlo to obtain 10^6 samples from $\Theta \mid X$, which takes about 30 minutes. Using these sample, the second estimator is the Monte Carlo estimate of the Minimum Mean Square Error (MC MMSE) estimator (i.e., the empirical average of the samples). The third *estimator, we use the sample based estimator of the mode described in [111] to compute a Monte Carlo estimate of the MAP (MC MAP).*

Our estimator $\theta^{\triangledown}$ is significantly closer to real θ and to the MCMC estimate of the MAP than the naive MAP. $\theta^{\triangledown}$ is also approximately as distant to the true value of θ as the MMSE estimate. Although none of them is the real MAP, these results suggest that $\theta^{\triangledown}$ accurately captures the estimation problem and provides a better result than naively trying to estimate $d_{1:T}$ as in the naive MAP.

2.3.3 Bayesian Optimal Experiment Design

The goal in experiment design is to find inputs for an estimation problem that will yield samples that provide "more information per sample". Consider a random vector X with *pdf* $p_X(x;u,\theta)$ where θ is a vector of unknown parameters and u a vector of control decision taking values in $\mathcal{U} \subset \mathbb{R}^N$. The Fisher Information Matrix is

$$\mathcal{FI}(u,\theta) = \mathbb{E}\left[\frac{\mathrm{d}\log p_X(X;u,\theta)}{\mathrm{d}\theta}\frac{\mathrm{d}\log p_X(X;u,\theta)'}{\mathrm{d}\theta}\right],$$

where the expected value is taken with respect to X and where we use the denominator-layout notation for the derivatives (producing column vectors). The Cramer-Rao lower bound states that, given any unbiased estimator $\hat{\theta}(u,X)$ of θ, its covariance

$$\mathbb{E}\Big[(\hat{\theta}(u,X)-\theta)(\hat{\theta}(u,X)-\theta)'\Big]$$

is lower bounded (in the positive definite matrix sense) by $\mathcal{FI}(u,\theta)^{-1}$. Therefore, if one minimizes (according to some criteria) $\mathcal{FI}(u,\theta)^{-1}$, one will decrease the covariance of any estimator achieving the Cramer-Rao bound.

In Bayesian optimal experiment design one assumes that θ is a realization of an underlying random vector Θ, with $pdf p_\Theta(\cdot)$, and select u^* to minimize the Bayesian

D-optimality (the D stands for determinant), criteria:

$$u^* \in \arg\min_{u \in \mathcal{U}} \mathbb{E}[\log\det\big(\mathcal{FI}(u,\Theta)^{-1}\big)], \tag{2.38}$$

where the expected value is taken with respect to Θ. It is shown in [12] that (2.38) optimizes the gain in the Shannon information of the experiment when $\hat{\theta}(u,X)$ is a Gaussian distribution with mean θ and covariance $\mathcal{FI}(u,\theta)^{-1}$. In other words, it designs an experiment that brings more information on average. Alternative Bayesian criteria include A-optimality (the A stands for average), where one wants to find a u^* such that

$$u^* \in \arg\min_{u \in \mathcal{U}} \mathbb{E}[\mathrm{tr}(\mathcal{FI}(u,\Theta)^{-1})]. \tag{2.39}$$

In this case, (2.39) minimizes the mean square error of any estimator $\hat{\theta}(u,X)$ that is unbiased and achieves the Cramer-Rao bound.

The experiment design in (2.38) and (2.39) is an unconstrained form of (2.11). Using the additive upper bound in (2.13) leads to

$$\begin{aligned} V^{\triangle} &= \min_{u \in \mathcal{U},\epsilon} \max_{\theta \in \Omega} -\log\det(\mathcal{FI}(u,\theta)) + \epsilon \log p_{\Theta}(\theta) + \epsilon \mathcal{H}_{\Theta} \\ V^{\triangledown} &= \max_{\epsilon} \min_{u \in \mathcal{U},\theta \in \Omega} -\log\det(\mathcal{FI}(u,\theta)) + \epsilon \log p_{\Theta}(\theta) + \epsilon \mathcal{H}_{\Theta}. \end{aligned} \tag{2.40}$$

For Bayesian A-optimality (2.39), we obtain

$$\begin{aligned} V^{\triangle} &= \min_{u \in \mathcal{U},\epsilon} \max_{\theta \in \Omega} \ \mathrm{tr}(\mathcal{FI}(u,\theta)^{-1}) + \epsilon \log p_{\Theta}(\theta) + \epsilon \mathcal{H}_{\Theta} \\ V^{\triangledown} &= \max_{\epsilon} \min_{u \in \mathcal{U},\theta \in \Omega} \ \mathrm{tr}(\mathcal{FI}(u,\theta)^{-1}) + \epsilon \log p_{\Theta}(\theta) + \epsilon \mathcal{H}_{\Theta}. \end{aligned} \tag{2.41}$$

Example 2.3.6 (Optimal trajectories for thermal air wind detection) *A glider is an air vehicle that flies without propellers, using only wind forces to change its altitude. In order to move up, a glider needs to estimate the location and intensity of the thermal vertical wind that would push it [112–114].*

Given an air column, a common model for the intensity of the vertical wind speed at position $z = (x, y)$ is

$$w(\bar{w}, \gamma, \bar{z}, z) = \bar{w}e^{-\gamma\|z-\bar{z}\|_2^2}$$

where $\bar{z} = (\bar{x}, \bar{y})$ denotes the position of the thermal center, $\bar{w}$ the wind speed at the thermal center and $1/\gamma$ the thermal radius. Our goal is to estimate the thermal parameters $\theta = (\bar{w}, \gamma, \bar{z})$ based on noisy measurements of the vertical air speed of the form

$$V_t = w(\bar{w}, \gamma, \bar{z}, z_t) + N_t$$

where z_t is the location where the measurement is taken and N_t are independent zero mean Gaussian distribution with variance σ^2. The probability density function for T measurements $v = (v_1, ..., v_T)$ is given by

$$p_V(v; \theta) = \frac{1}{(2\pi)^{T/2}\sigma^T} e^{-\frac{\sum_{t=1}^{T}(v_t - w_t)^2}{2\sigma^2}}$$

where $w_t = w(\bar{w}, \gamma, \bar{z}, z_t)$. The Fisher Information matrix associated to the estimation of θ is given by,

$$\begin{aligned}\mathcal{FI}(z_{1:T}, \theta) = \mathbb{E}\left[\frac{\mathrm{d}\log p_V(V;\theta)}{\mathrm{d}\theta}\frac{\mathrm{d}\log p_V(V;\theta)}{\mathrm{d}\theta}'\right] &= \frac{1}{\sigma^4}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{l=1}^{T}(V_t - w_t)(V_l - w_l)\frac{\mathrm{d}w_t}{\mathrm{d}\theta}\frac{\mathrm{d}w_l}{\mathrm{d}\theta}'\right] \\ &= \frac{1}{\sigma^2}\sum_{t=1}^{T}\frac{\mathrm{d}w_t}{\mathrm{d}\theta}\frac{\mathrm{d}w_t}{\mathrm{d}\theta}'\end{aligned}$$

where

$$\frac{\mathrm{d}w_t}{\mathrm{d}\theta} = \left[\frac{\partial w_t}{\partial \bar{w}}, \frac{\partial w_t}{\partial \gamma}, \frac{\partial w_t}{\partial \bar{z}}\right]' = e^{-\gamma\|z_t - \bar{z}\|^2}\left[1, -\bar{w}\|z_t - \bar{z}\|^2, -\bar{w}\gamma(z_t - \bar{z})'\right]'.$$

Given prior distributions on $\bar{w}$, γ and $\bar{z}$, we want to find the measurement points $z_1, z_2, \ldots, z_T$ that minimize (2.38) subject to the constraint that the distance between two consecutive z_t should be no larger than Δz. As the problem is rotationally symmetric, we fix the y coordinate of the first point to be 0.

We assign the following prior distributions. Both $\bar{w}$ and γ follow a Gamma distribution with parameters respectively $(\alpha_{\bar{w}}, \beta_{\bar{w}})$ and $(\alpha_\gamma, \beta_\gamma)$ and the thermal center $\bar{z}$ follows a zero mean Gaussian distribution with covariance $\Sigma_{\bar{z}}$.

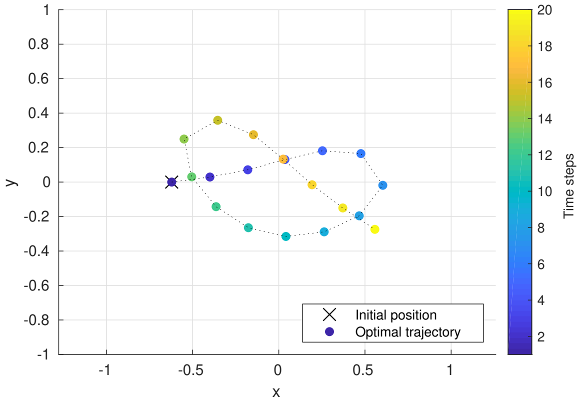


Figure 2.3: Optimal trajectory for the Bayesian experiment design for detecting the parameters of a vertical thermal air flow

We take the following numerical values. The number of measurements is $T = 20$*. The parameters of the priors are* $\alpha_{\bar{w}} = \alpha_\gamma = 1.25$*,* $\beta_{\bar{w}} = \beta_\gamma = 0.25$ *and* $\Sigma_{\bar{z}} = 0.1I$*. The maximum displacement between two sampling points is* $\Delta z = 0.05$*. The problem is highly nonconvex, requiring multiple initializations. For the lower bound, it takes about* 6.56 *seconds to run* 100 *optimizations with a random walk initialization, obtaining the lower bound* $V^\triangledown = -14.27$*. For the upper bound it takes about* 8.92 *seconds to run* 100 *optimizations with random walk initialization, obtaining the upper bound* $V^\vartriangle = 95.41$*. Using Monte Carlo integration, as discussed in Section 2.2.1, we obtain that the expected value of the log determinant of the Fisher Information Matrix given the trajectory is* 1.203*. The optimal trajectory can be seen in Figure 2.3.*

2.4 Conclusions and Future Work

We presented a general method to bound the expected value of any random variable with known probability density function. Stochastic programming is the main application of the bounds, where they can be used to determine an optimizer which has performance guarantees and satisfies inequality constraints. We illustrate the results with applications to finite-horizon stochastic control, estimation with latent variables and experiment design. The numerical results in theses applications show that optimizing the bound lead to solutions close to the optimal. They also suggest that even when the bounds are not tight, the argument that minimizes the upper bound is close to the one that minimizes the stochastic programming problem.

There are many future work directions to be considered. On the bounds themselves, most of the properties were determined for the additive and multiplicative bound, but other versions of the bounds could unlock other applications. The connection between the bounds we developed and distributionally robust optimization remains to be further understood, in particular for which kind of problem which approach is more suited. On obtaining solutions to the minmax optimization, an area for future research motivated by [23–25] arises from replacing the essential suprema used in the upper bound in (2.13) by maxima over independent samples of the random variable D and establishing sample complexity bounds to guarantee that the resulting optimization still provides an upper bound with high probability.

In terms of stochastic control, an evident extension would be stochastic model predictive control. In the estimation section, it would be interesting to study the asymptotic properties of the bound. As for new applications, machine learning is an area of significant potential. In particular, this method could either be used to accelerate the training of Neural Networks when there is a partial knowledge of the underlying model or in adversarial training.

2.A Appendix of Chapter 2

2.A.1 Penalization term for common distributions

Gaussian distribution The probability density function of a Gaussian Distribution with mean μ and covariance matrix Σ is

$$p_D(d) = \det(2\pi\Sigma)^{-1/2} \exp\left(-\frac{1}{2}\|d-\mu\|^2_{\Sigma^{-1}}\right)$$

where we use the notation $\|v\|^2_Q \coloneqq v'Qv$.

For the additive bound, $\mathcal{H}_D \coloneqq \mathbb{E}[-\log p_D(D)] = \frac{1}{2}\log\det(2\pi e\Sigma)$, therefore the penalization term simplifies to

$$\log p_D(d) + \mathcal{H}_D = -\frac{1}{2}\|d-\mu\|^2_{\Sigma^{-1}} + \frac{1}{2}M$$

where M is the dimension of D.

For the multiplicative bound, $\mathcal{I}_D(\epsilon) \coloneqq \mathbb{E}[p_D(D)^{-\epsilon}] = \det(2\pi\Sigma)^{\epsilon/2}(1-\epsilon)^{-M/2}$ if $\epsilon < 1$ and $+\infty$ otherwise, therefore for $\epsilon < 1$ the penalization terms simplifies to

$$p_D(d)^{\epsilon}\mathcal{I}_D(\epsilon) = \exp\left(-\frac{1}{2}\epsilon\|d-\mu\|^2_{\Sigma^{-1}}\right)(1-\epsilon)^{-M/2}$$

.

Uniform distribution If D is a Uniform distribution over a bounded support $\mathcal{D}$, its *pdf*is $p_D(d) = \mathcal{V}_{\mathcal{D}}^{-1}\mathbb{1}_{\mathcal{D}}(d)$ where $\mathbb{1}_{\mathcal{D}}(\cdot)$ is the indicator function of $\mathcal{D}$ and $\mathcal{V}_{\mathcal{D}} = \mathbb{E}[\mathbb{1}_{\mathcal{D}}(D)]$ is the volume of $\mathcal{D}$.

For the additive bound, $\mathcal{H}_D = \mathbb{E}[-\log(\mathbb{1}_{\mathcal{D}}(D))] + \log(\mathcal{V}_{\mathcal{D}}) = \mathbb{E}[0] + \log(\mathcal{V}_{\mathcal{D}})$, therefore the penalization terms simplifies to $\log p_D(d) + \mathcal{H}_D = 0 \ \forall d \in \mathcal{D}$.

For the multiplicative bound, $\mathcal{I}_D(\epsilon) = \mathbb{E}[(\mathcal{V}_{\mathcal{D}})^{\epsilon}\mathbb{1}_{\mathcal{D}}(D)^{-\epsilon}] = (\mathcal{V}_{\mathcal{D}})^{\epsilon}$, therefore the penalization term simplifies to $p_D(d)^{\epsilon}\mathcal{I}_D(\epsilon) = 1 \ \forall d \in \mathcal{D}$.

2.A.2 Proofs of Section 2.1

To prove the results that follow, we need the following properties of the essential supremum and infimum which we state without a proof.

Lemma 2.A.1 *Given two random variables X and Y then*

$$
\begin{aligned}
&X \overset{\text{wpo}}{\geqslant} Y \quad \Rightarrow \quad \text{ess inf}\, X \geqslant \text{ess inf}\, Y \\
&X \overset{\text{wpo}}{\geqslant} Y \quad \Rightarrow \quad \text{ess sup}\, X \geqslant \text{ess sup}\, Y \\
&\text{ess inf}(X+Y) \geqslant \text{ess inf}\, X + \text{ess inf}\, Y \\
&\text{ess sup}(X+Y) \leqslant \text{ess sup}\, X + \text{ess sup}\, Y
\end{aligned}
\qquad \square
$$

Proof of Proposition 2.1.4. Take $\epsilon_1, \epsilon_2 \in \mathbb{R}$ such that $J^*(\epsilon_1)$, $J^*(\epsilon_2) < +\infty$ and $\lambda \in [0,1]$

$$
\begin{aligned}
&J^*(\lambda\,\epsilon_1 + (1-\lambda)\epsilon_2) \\
&= \text{ess sup}\, V(D) + (\epsilon_1\lambda + \epsilon_2(1-\lambda)) \log p_D(D) \\
&= \text{ess sup}(\lambda + 1 - \lambda)V(D) + (\epsilon_1\lambda + \epsilon_2(1-\lambda)) \log p_D(D) \\
&\leqslant \lambda\, \text{ess sup}\, V(D) + \epsilon_1 \log p_D(D) \\
&\qquad + (1-\lambda)\, \text{ess sup}\, V(D) + \epsilon_2 \log p_D(D) < \infty
\end{aligned}
$$

where the inequality follows from Lemma 2.A.1. This establishes that the additive upper bound is convex in ϵ and that $J^*(\epsilon)$ is finite on a convex set.

For the multiplicative bound, it remains to show that $\mathcal{I}_D(\epsilon)$ is log convex: take $\epsilon_1, \epsilon_2 \in \mathbb{R}$ such that $\mathcal{I}_D(\epsilon_1), \mathcal{I}_D(\epsilon_2)$ are finite and $\lambda \in [0,1]$. By applying Hölder's inequality we obtain

$$
\begin{aligned}
&\mathbb{E}[p_D(D)^{-\lambda\epsilon_1} p_D(D)^{-(1-\lambda)\epsilon_2}] \\
&\leqslant \Big(\mathbb{E}[p_D(D)^{-\lambda\epsilon_1/\lambda}]\Big)^{\lambda} \Big(\mathbb{E}[p_D(D)^{-(1-\lambda)\epsilon_2/(1-\lambda)}]\Big)^{(1-\lambda)}
\end{aligned}
$$

$$=\left(\mathbb{E}[p_D(D)^{-\epsilon_1}]\right)^{\lambda}\left(\mathbb{E}[p_D(D)^{-\epsilon_2}]\right)^{(1-\lambda)}$$

which establishes log convexity. ■

2.A.3 Necessary and sufficient conditions for finite bounds

Consider a constant $\gamma > 0$ sufficiently small so that $\mathrm{P}\left(p_D(D) > \gamma\right) > 0$. We say a measurable function $f(\cdot)$ is γ-essentially upper bounded if

$$\text{ess sup}\left[f(D) \mid p_D(D) > \gamma\right] < \infty,$$

γ-essentially lower bounded if

$$\text{ess inf}\left[f(D) \mid p_D(D) > \gamma\right] > -\infty,$$

and γ-essentially bounded if it is both γ-essentially upper and lower bounded.

Theorem 2.A.2 (Finite bounds) *Suppose that $p_D(\cdot)$ is γ-essentially upper bounded and let $c \in (0, 1/\gamma)$ be any constant for which*

$$\text{ess sup}\left[p_D(D) \mid p_D(D) > \gamma\right] \leqslant 1/c, \tag{2.42}$$

and ϵ an arbitrary finite constant. Regarding the additive bound: Assuming that $V(\cdot)$ is γ-essentially lower bounded, then

$$p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \text{ess inf}\left[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\right] > \epsilon$$
$$\Rightarrow \quad \text{ess inf}\left(V(D) + \epsilon \log p_D(D)\right) > -\infty. \tag{2.43}$$

Conversely,

$$\text{ess inf}\left(V(D) + \epsilon \log p_D(D)\right) > -\infty$$
$$\Rightarrow \quad p_D(D) \overset{\text{wpo}}{>} \gamma \text{ or } \exists L > 0 :$$

$$\text{ess inf}\Big[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] \geqslant \epsilon + \frac{L}{\log c\gamma}. \quad (2.44)$$

Assuming that $V(\cdot)$ γ-essentially upper bounded, then

$$\begin{aligned} &p_D(D) \overset{\text{wpo}}{>} \gamma \textit{ or } \text{ess sup}\Big[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] < \epsilon \\ &\qquad\Rightarrow \quad \text{ess sup}\big(V(D) + \epsilon \log p_D(D)\big) < +\infty \quad (2.45)\end{aligned}$$

Conversely,

$$\begin{aligned} &\text{ess sup}\big(V(D) + \epsilon \log p_D(D)\big) < \infty \\ &\qquad\Rightarrow \quad p_D(D) \overset{\text{wpo}}{>} \gamma \textit{ or } \exists L > 0 : \\ &\qquad\qquad \text{ess sup}\Big[\frac{-V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] \leqslant \epsilon - \frac{L}{\log c\gamma}. \quad (2.46)\end{aligned}$$

Regarding the multiplicative bound: Assuming that $\log V(\cdot)$ *is γ-essentially lower bounded, then*

$$\begin{aligned} &p_D(D) \overset{\text{wpo}}{>} \gamma \textit{ or } \text{ess inf}\Big[\frac{-\log V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] > \epsilon \\ &\qquad\Rightarrow \quad \text{ess inf}\big(\log V(D) + \epsilon \log p_D(D)\big) > -\infty.\end{aligned}$$

Conversely,

$$\begin{aligned} &\text{ess inf}\big(\log V(D) + \epsilon \log p_D(D)\big) > -\infty \\ &\qquad\Rightarrow \quad p_D(D) \overset{\text{wpo}}{>} \gamma \textit{ or } \exists L > 0 : \\ &\qquad\qquad \text{ess inf}\Big[\frac{-\log V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] \geqslant \epsilon + \frac{L}{\log c\gamma}. \quad (2.47)\end{aligned}$$

Assuming that $\log V(\cdot)$ *is γ-essentially upper bounded, then*

$$\begin{aligned} &p_D(D) \overset{\text{wpo}}{>} \gamma \textit{ or } \text{ess sup}\Big[\frac{-\log V(D)}{\log c\, p_D(D)} \mid p_D(D) \leqslant \gamma\Big] < \epsilon \\ &\qquad\Rightarrow \quad \text{ess sup}\big(\log V(D) + \epsilon \log p_D(D)\big) < +\infty\end{aligned}$$

Conversely,

$$\begin{aligned}\text{ess sup}\,\big(\log V(D)+\epsilon\log p_D(D)\big)<\infty&\\ \Rightarrow\quad p_D(D)\overset{\text{wpo}}{>}\gamma\ \textit{or}\ \exists L>0:&\\ \text{ess sup}\Big[\frac{-\log V(D)}{\log c\,p_D(D)}\;\big|\;p_D(D)\leqslant\gamma\Big]\leqslant\epsilon-\frac{L}{\log c\gamma}.&\qquad(2.48)\end{aligned}$$

□

Proof. We will prove the theorem for the additive lower bound [*i.e.*, (2.43) and (2.44)]. The proof for the other bounds can be obtained in an analogous way.

To prove (2.43), we note that since $V(\cdot)$ is γ-essentially lower bounded there exists a finite constant L such that

$$\text{ess inf}\,\big[V(D)\mid p_D(D)>\gamma\big]\geqslant L.$$

In view of this and (2.42), we have that

$$\mathrm{P}\,\big(V(D)\geqslant L, p_D(D)\leqslant 1/c\mid p_D(D)>\gamma\big)=1. \qquad (2.49)$$

Since

$$V(D)\overset{\text{wpo}}{\geqslant}L,\ p_D(D)\overset{\text{wpo}}{\leqslant}1/c,\ p_D(D)\overset{\text{wpo}}{>}\gamma\quad\Rightarrow\quad V(D)+\epsilon\log c\,p_D(D)\overset{\text{wpo}}{\geqslant}L^*>-\infty,$$

with $L^*\coloneqq L-|\epsilon|\,|\log c\gamma|$, we conclude from (2.49) that

$$\mathrm{P}\,\big(V(D)+\epsilon\log c\,p_D(D)\geqslant L^*\;\big|\;p_D(D)>\gamma\big)=1. \qquad (2.50)$$

In case $p_D(D)\overset{\text{wpo}}{>}\gamma$, we conclude that the corresponding unconditional probability satisfies the same bound and (2.43) follows. Otherwise,

$$\text{ess inf}\Big[\frac{-V(D)}{\log c\,p_D(D)}\;\big|\;p_D(D)\leqslant\gamma\Big]>\epsilon$$

implies that

$$\mathrm{P}\left(\frac{-V(D)}{\log c\, p_D(D)} \geqslant \epsilon \;\middle|\; p_D(D) \leqslant \gamma\right) = 1. \tag{2.51}$$

Since

$$p_D(D) \leqslant \gamma \;\overset{\text{wpo}}{\Longrightarrow}\; \log c\, p_D(D) \leqslant \log c\, \gamma < 0, \tag{2.52}$$

we also conclude from (2.51) that

$$\mathrm{P}\left(V(D) + \epsilon \log c\, p_D(D) \geqslant 0 \;\middle|\; p_D(D) \leqslant \gamma\right) = 1. \tag{2.53}$$

Combining (2.53) and (2.50), we conclude that the corresponding unconditional probability satisfies

$$\mathrm{P}\left(V(D) + \epsilon \log c\, p_D(D) \geqslant \min\{0, L*\}\right) = 1,$$

from which (2.43) follows.

To prove (2.44), we use the fact that ess inf $\left(V(D) + \epsilon \log p_D(D)\right) > -\infty$ implies that there exists some finite scalar $L > 0$, for which

$$\mathrm{P}(V(D) + \epsilon \log c\, p_D(D) \geqslant -L) = 1. \tag{2.54}$$

When $p_D(D) \overset{\text{wpo}}{>} \gamma$ the implication in (2.44) is tautologically true, so we focus our attention on the case $\mathrm{P}(p_D(D) \leqslant \gamma) > 0$, for which (2.54) implies that

$$\mathrm{P}\left(V(D) + \epsilon \log c\, p_D(D)\right) \geqslant -L \;\middle|\; p_D(D) \leqslant \gamma\Big) = 1. \tag{2.55}$$

Using (2.52), we conclude that

$$V(D) + \epsilon \log c\, p_D(D),\; p_D(D) \leqslant \gamma \quad \overset{\text{wpo}}{\Rightarrow} \quad \frac{-V(D)}{\log c\, p_D(D)} \geqslant \epsilon + \frac{L}{\log c\, \gamma}$$

and therefore (2.55) implies that

$$\mathrm{P}\left(\frac{-V(D)}{\log c\, p_D(D)} \geqslant \epsilon + \frac{L}{\log c\, \gamma} \;\middle|\; p_D(D) \leqslant \gamma\right) = 1. \tag{2.56}$$

This shows that

$$\operatorname{ess\,inf}\left[\frac{-V(D)}{\log c\, p_D(D)} \,\Big|\, p_D(D) \leqslant \gamma\right] \geqslant \epsilon + \frac{L}{\log c\gamma},$$

which completes the proof of the implication in (2.44). ■

Chapter 3

Newton and interior-point methods for (constrained) nonconvex-nonconcave minmax optimization with stability guarantees

Parts of this chapter come from [1]

In this chapter, we consider the problem of using second order methods to find local minmax point. Second order methods are crucial for real time applications, such as robust model predictive control and robust estimation, where using first order methods tends to be unfeasible due to their slow convergence. However, for nonconvex-nonconcave problems, second order methods can converge to a point that is not a local minmax.

In order to better explain our approach for minmax optimization, we start by presenting related results for minimization, in Section 3.1. First, we present a widely known modification of Newton method used in nonconvex minimization. We show that this modification can be obtain by requiring the minimization of a second order approximation of the function to be strongly convex. We then show that a property not well know about this modification is that it guarantees that the only locally asymptotic stable equilibrium points of the Newton method iterations are the local minimum points. In essence, the property guarantee that the modified Newton iteration can only converge to a local minima. Building on these results, we then turn our attention to constrained minimization. We start by presenting a modified interior-point method and show how each descent direction can be obtained from the minimization of a quadratic minimization with linear constraints. Similar to the case of unconstrained minimization, we show that this modified interior-point method is such that the only stable equilibrium points are local minima.

Based on these results, we address the main topic of the chapter, *i.e.*, the construction of second order methods for minmax optimization, in Section 3.2. Inspired by the modification described for minimization case, we construct a modified Newton which guarantees that the minmax optimization of a second order approximation of the function is well defined. We then show, using counter example, that these conditions are not enough to guarantee that the only locally stable equilibrium points are local minmax. We then derive the conditions such that the modified Newton method guarantees that the only locally stable equilibrium points of the Newton iterations are local minmax. We then extend these result by developing an interior-point methods for constrained minmax and deduce what are the conditions to obtain the appropriate stability results.

Finally, in Section 3.3 we describe how our results can be implemented in an algorithm to find local minmax points. We test our algorithm in some benchmark examples to show their efficiency. In addition, we test the algorithm in the homicidal chauffeur problem, and show that if the Hessian matrix is sparse, the time to solve the optimization scales roughly linearly with the number of nonzero elements in the Hessian. The property is usually found in many problems where the cost function can be represented using stages.

Notation: The set of real numbers is denoted by $\mathbb{R}$. Given a vector $v \in \mathbb{R}^n$, its transpose is denoted by v'. The operation $\operatorname{diag}(v)$ creates a matrix with diagonal elements v and off-diagonal elements 0. The matrix I is the identity, $\mathbf{1}$ is the matrix of ones and $\mathbf{0}$ the matrix of zeros; their sizes will be provided as subscripts whenever it is not clear from context. If a matrix A only has real eigenvalues, we denote by $\lambda_{min}(A)$ and $\lambda_{max}(A)$ its smallest and largest eigenvalues. The inertia of A is denoted by $\operatorname{In}(A)$, and is a 3-tuple with the number of positive, negative and zero eigenvalues of A.

Consider a differentiable function $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^p$. The Jacobian (or gradient if $p = 1$) at a point $(\bar{x}, \bar{y})$ according to the x variable is a matrix of size $n \times p$ and is denoted by $\boldsymbol{\nabla}_x f(\bar{x}, \bar{y})$, and analogously for the variable y. When $p = 1$ and $f(\cdot)$ is twice differentiable, we use the notation $\boldsymbol{\nabla}_{yx} f(\bar{x}, \bar{y}) := \boldsymbol{\nabla}_y \big(\boldsymbol{\nabla}_x f\big)(\bar{x}, \bar{y})$ which has sizes $m \times n$. We use analogous definition for $\boldsymbol{\nabla}_{xy} f(\bar{x}, \bar{y})$, $\boldsymbol{\nabla}_{xx} f(\bar{x}, \bar{y})$ and $\boldsymbol{\nabla}_{yy} f(\bar{x}, \bar{y})$.

3.1 Minimization

Let $f : \mathcal{X} \to \mathbb{R}$ be a twice continuously differentiable cost function defined in a set $\mathcal{X} \subset \mathbb{R}^{n_x}$, and consider the minimization problem

$$\min_{x \in \mathcal{X}} f(x). \tag{3.1}$$

We recall that a point x^* is called a local minimum of $f(\cdot)$ if there exist $\delta > 0$ such that $f(x^*) \leqslant f(x)$ for all $x \in \{x \in \mathcal{X} : \|x - x^*\| < \delta\}$. We will study the property of Newton type algorithms to solve (3.1) in two distinct cases, when $\mathcal{X} = \mathbb{R}^{n_x}$ and when $\mathcal{X}$ is defined by equality and inequality constraints.

3.1.1 Unconstrained minimization

Let $\mathcal{X} = \mathbb{R}^{n_x}$, which is referred to as unconstrained minimization in the literature. If $f(\cdot)$ is twice continuously differentiable in a neighborhood of a point x and $\boldsymbol{\nabla}_x f(x) = \mathbf{0}$ and $\boldsymbol{\nabla}_{xx} f(x) \succ 0$, then x is a local minimum of $f(\cdot)$ [54, Chapter 2].

An extremely popular method to solve a minimization problem is to use Newton's root finding method to obtain a point x such that $\boldsymbol{\nabla}_x f(x) = \mathbf{0}$. In its most basic form, the algorithm's iterations are given by

$$x^+ = x + d_x = x - \boldsymbol{\nabla}_{xx} f(x)^{-1} \boldsymbol{\nabla}_x f(x). \tag{3.2}$$

where we use the notation x^+ to designate the value of x at the next iteration. Newton's method biggest advantage is that it converges very fast near any point that satisfies the first order condition $\boldsymbol{\nabla}_x f(x) = \mathbf{0}$: at least linearly but possibly superlinearly when the function is Lipschitz [54, Theorem 3.6]. However, this is also precisely Newton's method biggest limitation for nonconvex minimization, because it does not distinguish a local minimum from any other point satisfying the first order condition. Let us further illustrate this limitation with an example.

Example 3.1.1 *Consider the optimization,*

$$\min_{x \in \mathbb{R}} x^3 - 3x, \tag{3.3}$$

for which $\forall x \in \mathbb{R}$,

$$f(x) := x^3 - 3x, \qquad \boldsymbol{\nabla}_x f(x) = 3x^2 - 3, \qquad \boldsymbol{\nabla}_{xx} f(x) = 6x.$$

The corresponding Newton iteration (3.2) *is of the form*

$$x^+ = x - \frac{3x^2 - 3}{6x},$$

for which both the local minimum $x^{\min} := 1$ and the local maximum $x^{\max} := -1$ are locally asymptotically stable equilibria with superlinear convergence. Specifically,

$$\begin{cases} x_0 > 0 \Rightarrow x_k \to x^{\min} := 1, \textit{(local minimum)}, \\ x_0 < 0 \Rightarrow x_k \to x^{\max} := -1, \textit{(local maximum)}, \\ x_0 = 0 \Rightarrow \textit{iteration fails since } \boldsymbol{\nabla}_{xx} f(x) = 6x \textit{ is not invertible.} \end{cases}$$

Moreover, the iteration never actually "converges" to the global "infimum" $x \to -\infty$.

In order to address this limitation, a widely used modification of Newton's method for unconstrained nonconvex optimization [54, Chapter 3.4], is obtained by modifying the basic Newton method such that d_x is obtained from solving the following local quadratic approximation to (3.1)

$$\begin{aligned} d_x &= \arg\min_{\bar{d}_x} f(x) + \boldsymbol{\nabla}_x f(x)'\bar{d}_x + \frac{1}{2}\bar{d}_x(\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I)\bar{d}_x \\ &= -(\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_x f(x) \end{aligned} \tag{3.4}$$

with $\epsilon_x(x) \geqslant 0$ chosen such that $(\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I)$ is positive definite. For twice differentiable strongly-convex functions we can choose $\epsilon_x(x) = 0$ and this corresponds to the classical Newton's method. However, when $f(\cdot)$ is not strongly-convex, the minimization in (3.4) is only well-defined if $\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I$ is positive definite, which requires selecting a strictly positive value for $\epsilon_x(x)$, leading to a modified Newton's method. Regardless of whether $f(\cdot)$ is convex, the positive definiteness of $\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I$ guarantees that $d_x'\boldsymbol{\nabla}_x f(x) = -\boldsymbol{\nabla}_x f(x)(\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_x f(x) < 0$ and therefore d_x is a descent direction at x [54]. The corresponding Newton iteration to obtain a local minimum is then given by

$$x^+ = x + d_x = x - (\boldsymbol{\nabla}_{xx} f(x) + \epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_x f(x). \tag{3.5}$$

Let us analyze how this modification impacts the convergence in our previous example.

Example 3.1.2 (Continuation) *For the optimization in* (3.3)*, the modified Newton step in* (3.5) *becomes* $x^+ = x - \frac{3x^2-3}{6x+\epsilon_x(x)}$ *with* $\epsilon_x(\cdot)$ *such that*

$$\begin{cases} \epsilon_x(x) \geqslant 0 & x > 0, \\ \epsilon_x(x) > -6x & x \leqslant 0. \end{cases} \tag{3.6}$$

In this case,

$$\begin{cases} x_0 > x^{\max} := -1 \Rightarrow & x_k \to x^{\min} := 1 \textit{(local minimum)}, \\ x_0 < x^{\max} := -1 \Rightarrow & x_k \to -\infty \textit{ (global “infimum”)}, \\ x_0 = x^{\max} := -1 \Rightarrow & x_k = x^{\max}, \forall k \textit{(unstable equilibrium)}. \end{cases}$$

Selecting the function $\epsilon_x(\cdot)$ *with* $\epsilon_x(\cdot) = 0$ *around* $x^{\min}$ *results in superlinear convergence to* $x^{\min}$*, but if* $\epsilon_x(\cdot) > 0$*, the convergence is only linear. For example, picking* $\epsilon_x(x) = -6x + \eta$ *with* $\eta > 0$*,* (3.6) *holds for all* x*, but the modified Newton step in* (3.5) *becomes* $x^+ = x - \frac{3x^2-3}{\eta}$*, which is just a gradient descent.*

The following result generalizes the conclusion from the previous example by establishing that the positive definiteness of $\nabla_{xx} f(x) + \epsilon_x(x) I$ not only guarantees that d_x is a descent direction, but also that every locally asymptotically stable (LAS) equilibrium point of the Newton iteration (3.5) is a local minimum.

Theorem 3.1.3 (Stability of modified Newton method for minimization) *Let* x *be an equilibrium point in the sense that* $\nabla_x f(x) = \mathbf{0}$*. Assume that* $\nabla_{xx} f(x)$ *is invertible and that* $\nabla_{xx} f(\cdot)$ *is differentiable in a neighborhood around* x*. Then for any function* $\epsilon_x(\cdot)$ *that is constant in a neighborhood around* x *and satisfies* $\nabla_{xx} f(x) + \epsilon_x(x) I \succ \mathbf{0}$ *one has that if:*

i) x *is a local minimum, then it is a LAS equilibrium point of* (3.5)*.*

ii) x is not a local minimum, then it is an unstable equilibrium point of (3.5).

The theorem's first implication is that if the modified Newton iteration starts sufficiently close to a strict local minimum, it will converge at least linearly fast to it. One could think that it would always be preferable to have $\epsilon_x(x) = 0$ if $\nabla_{xx} f(x) \succ 0$, in which case not only stability can be trivially obtained but also that the Newton method has superlinear convergence if $f(\cdot)$ is Lipschitz [54, Theorem 3.6]. However, in practice, there are situations for which one might want to take $\epsilon_x(x) > 0$. A typical case happens if the smallest eigenvalue of $\nabla_{xx} f(x)$ is positive but very small, which might bring numerical issues when computing the Newton step $\nabla_{xx} f(x)^{-1} \nabla_x f(x)$. This issue can be fixed by taking $\epsilon_x(x) > 0$, and Theorem 3.1.3 guarantees that doing so will not impair (at least locally) the algorithm's capacity to converge towards a local minimum.

The theorem's second implication is, in a way, even more relevant than the first one. As we mentioned earlier, the regular Newton's method (meaning, with $\epsilon_x(x) = 0$) is infamously known to be attracted to any point that satisfies $\nabla_x f(x) = \mathbf{0}$, regardless of whether it is a local minimum, a saddle point, or a local maximum. What Theorem 3.1.3 is essentially saying is that the modified Newton is only attracted to local minima, and that any other equilibrium point repels the iteration. In essence, this means that the modified Newton's method cannot converge towards a point that is not a local minimum, thus fixing one of the biggest drawbacks of the regular Newton's method.

Proof of Theorem 3.1.3. From our assumption that $\nabla_{xx} f(x)$ is invertible, x is a local minimum if and only if $\nabla_{xx} f(x) \succ 0$. This comes from the second order necessary condition for minimization [54, Chapter 2].

Let us now prove the stability and instability properties. The first step in our analysis is to calculate the Jacobian of $(\nabla_{xx} f(x) + \epsilon_x(x) I)^{-1} \nabla_x f(x)$ that appears in (3.5) at an equilibrium point x. Using the differentiability of $\nabla_{xx} f(\cdot)$ and that $\epsilon_x(\cdot)$ is constant in a neighborhood of x, we obtain that

$$\begin{aligned}\boldsymbol{\nabla}_x\Big((\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_x f(x)\Big) &= (\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_{xx}f(x)+\\ &\qquad\sum_{i=1}^{N}\boldsymbol{\nabla}_x[(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}]_i\boldsymbol{\nabla}_x f(x)^{(i)}\end{aligned}$$

where $\boldsymbol{\nabla}_x f(x)^{(i)}$ is the i$^{\text{th}}$ element of $\boldsymbol{\nabla}_x f(x)$ and $[(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}]_i$ is the i$^{\text{th}}$ column of $(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}$. Since $(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)$ is positive definite, $\boldsymbol{\nabla}_x[(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}]_i$ is well defined and since x is an equilibrium point, $\boldsymbol{\nabla}_x f(x)^{(i)} = 0$ for $i \in \{1\ldots N\}$ and therefore the Jacobian of right-hand side of (3.5) is given by

$$\boldsymbol{\nabla}_x\Big(x-(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_x f(x)\Big) = I-(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_{xx}f(x). \tag{3.7}$$

The main argument of the proof is based on the following result. Let v be an eigenvector associated to an eigenvalue ρ of (3.7). Then

$$\begin{aligned}&\Big(I-(\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_{xx}f(x)\Big)v=\rho v\\ &\qquad\Leftrightarrow (1-\rho)v = (\boldsymbol{\nabla}_{xx}f(x)+\epsilon_x(x)I)^{-1}\boldsymbol{\nabla}_{xx}f(x)v\\ &\Leftrightarrow \Big(\rho\boldsymbol{\nabla}_{xx}f(x)+(\rho-1)\epsilon_x(x)I\Big)v=\mathbf{0}\end{aligned} \tag{3.8}$$

Therefore, ρ is an eigenvalue of (3.7) if and only if $\rho\boldsymbol{\nabla}_{xx}f(x)+(\rho-1)\epsilon_x(x)I$ is singular.

We remind the reader that given a dynamical system, if the system's dynamic equation is continuously differentiable, a point is a LAS equilibrium point if all the eigenvalues of the linearized system are inside the unit circle. Conversely, if at least one of the eigenvalues of the linearized system is outside the unit circle, then the system is unstable [115, Chapter 8].

From (3.8), $\rho = 0$ is an eigenvalue if and only if $\epsilon_x(x) = 0$, which, by construction, can only happen if x is a local minimum, in which case x is a LAS equilibrium point of (3.5), as expected.

For $\rho \neq 0$, let us rewrite this expression as $\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)I$ with $\mu := 1 - 1/\rho$. We conclude that x is a LAS equilibrium point of (3.5) if $\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)$ is nonsingular $\forall \mu \in [0, 2]$. Conversely, x is an unstable equilibrium point of (3.5) if $\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)$ is singular for some $\mu \in [0, 2]$.

If x is a local minimum, then $\lambda_{min}(\boldsymbol{\nabla}_{xx} f(x)) > 0$. As $\epsilon_x(x) > 0$, we conclude that $\lambda_{min}(\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)I) > 0$ for every $\mu \geqslant 0$ and therefore x is a LAS equilibrium point of (3.5). Conversely, if x is not a local minimum then $\lambda_{min}(\boldsymbol{\nabla}_{xx} f(x)) < 0$. By construction of $\epsilon_x(x)$, we have that $\lambda_{min}(\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)I) > 0$, which, by continuity of the eigenvalue, implies $\exists \mu \in (0, 1)$ such that $\lambda_{min}(\boldsymbol{\nabla}_{xx} f(x) + \mu\epsilon_x(x)I) = 0$. Therefore x is an unstable equilibrium point of (3.5).

3.1.2 Constrained minimization

Our results from the previous section can also be extended to consider the case with more general constraint with the minimization set $\mathcal{X}$ involving equality and inequality constraints of the form

$$\mathcal{X} = \{x \in \mathbb{R}^n : G_x(x) = \mathbf{0}, F_x(x) \leqslant \mathbf{0}\}$$

where the functions $G_x : \mathbb{R}^{n_x} \to \mathbb{R}^{l_x}$ and $F_x : \mathbb{R}^{n_x} \to \mathbb{R}^{m_x}$ are all twice continuously differentiable. It will be convenient for the development of the primal-dual interior-point method to use slack variables and rewrite (3.1) as

$$\min_{x, s_x : G_x(x) = \mathbf{0}, F_x(x) + s_x = \mathbf{0}, s_x \geqslant \mathbf{0}} f(x). \tag{3.9}$$

where $s_x \in \mathbb{R}^{m_x}$.

Similar to what we have in the unconstrained minimization, we want a second order conditions to determine whether a point is a local minimum. Consider the function

$$L(z) = f(x) + \nu_x' G_x(x) + \lambda_x'(F_x(x) + s_x),$$

where we use the shorthand notation $z := (x, s_x, \nu_x, \lambda_x)$. $L(z)$ is essentially the Lagrangian of (3.9). In order to present the second order conditions, we need to define two concepts, the linear independence constraint qualification and strict complementarity [54, Definitions 12.4 and 12.5].

Definition 3.1.4 (LICQ and strict complementarity) *Let the sets of active inequality constraints for the minimization be defined by*

$$\mathcal{A}_x(x) = \{i : F_x^{(i)}(x) = 0, i = 1, \dots, m_x\}$$

where $F_x^{(i)}(x)$ denote the i^{th} element of $F_x(x)$. Then:

- *The linear independence constraint qualification (LICQ) is said to hold at z if the vectors in the set*

  $$\{\boldsymbol{\nabla}_x G_x^{(i)}(x), i = 1, \dots, l_x\} \bigcup \{\boldsymbol{\nabla}_x F_x^{(i)}(x), i \in \mathcal{A}_x(x)\}$$

  are linearly independent.

- *Strict complementarity is said to hold at x if $\lambda_x^{(i)} > 0 \ \forall i \in \mathcal{A}_x(x)$*

We have almost all the ingredients to present the second order condition for constrained minimization. For unconstrained minimization, a sufficient condition for a point x to be a local minimum is that $\boldsymbol{\nabla}_x f(x) = 0$ and $\boldsymbol{\nabla}_{xx} f(x) \succ 0$. If it were not for the inequality constraints in (3.9), we would be able to state the second order conditions using gradients and Hessians of $L(z)$. The inequality constraints make the statement a bit more complicated. The role of the gradient will be played by

$$g(z, b) := \begin{bmatrix} \boldsymbol{\nabla}_x L(z) \\ \lambda_x \odot s_x - b\mathbf{1} \\ G_x(x) \\ F_x(x) + s_x \end{bmatrix} \tag{3.10}$$

with $\odot$ denoting the element wise Hadamard product of two vectors and $b \geqslant 0$ the barrier parameter (its role will be explained shortly). The role of $\boldsymbol{\nabla}_{xx} f(x)$ in the unconstrained minimization will be played by the matrix

$$H_{zz} f(z) = \begin{bmatrix} \boldsymbol{\nabla}_{xx} L(z) & 0 & \boldsymbol{\nabla}_x G_x(x) & \boldsymbol{\nabla}_x F_x(x) \\ 0 & \operatorname{diag}(\lambda_x) & 0 & \operatorname{diag}(s_x^{1/2}) \\ \boldsymbol{\nabla}_x G_x(x)' & 0 & 0 & 0 \\ \boldsymbol{\nabla}_x F_x(x)' & \operatorname{diag}(s_x^{1/2}) & 0 & 0 \end{bmatrix}. \tag{3.11}$$

We also remind the reader that the inertia $\operatorname{In}(A)$ of a symmetric matrix A is a 3-tuple with the number of positive, negative and zero eigenvalues of A.

Proposition 3.1.5 (Second order conditions for constrained minimization) *Let z be an equilibrium point in the sense that $g(z, 0) = \mathbf{0}$ with $\lambda_x, s_x \geqslant \mathbf{0}$. If the LICQ and strict complementarity hold at z and*

$$\operatorname{In}(H_{zz} f(z)) = (n_x + m_x, l_x + m_x, 0) \tag{3.12}$$

then x is a local minimum of (3.9).

While this result is relatively well known, we present its proof in Appendix 3.A. The proof also makes it easier to understand the proof of the second order sufficient conditions for constrained minmax optimization.

Primal-dual interior-point method

Let $d_z := (d_x, d_s, d_\nu, d_\lambda)$ be the update direction for z, which will play an equivalent role to d_x in the unconstrained case. A basic primal-dual interior-point method finds a candidate solution to (3.9) using the iterations

$$z^+ = z + \alpha d_z = z - \alpha \boldsymbol{\nabla}_z g(z, b)'^{\,-1} g(z, b) \tag{3.13}$$

where the barrier parameter b is slowly decreased to 0, so that z converges to a root of $g(z,0) = \mathbf{0}$ while $\alpha \in (0,1]$ is chosen at each step such that the feasibility condition $\lambda_x, s_x > \mathbf{0}$ hold [54, Chapter 19]. This basic primal-dual interior-point has similar limitation as a (non-modified) Newton method for unconstrained minimization: it might converge towards an equilibrium point that is not a local minimum and $\boldsymbol{\nabla}_z g(z,b)$ might not be invertible. Similar to what we have done in the unconstrained case, we can modify this basic primal-dual interior-point method such that the update direction d_z is obtained from a quadratic program that locally approximates (3.9). The rest of this section will be spent mostly constructing such quadratic program.

Let us start with $\mathcal{X}$ described only by equality constraints (*i.e.* no $F_x(x)$ and no s_x), in which case $L(z) = f(x) + \nu_x' G_x(x)$. Consider the optimization

$$\min_{\bar{d}_x : G_x(x) + \boldsymbol{\nabla}_x G_x(x)' \bar{d}_x = \mathbf{0}} L(z) + \bar{d}_x' \boldsymbol{\nabla}_x L(z) + \frac{1}{2} \bar{d}_x' (\boldsymbol{\nabla}_{xx} L(z) + \epsilon_x(z) I) \bar{d}_x, \tag{3.14}$$

which locally approximates (3.9) around (x, ν_x) [1]. If $\boldsymbol{\nabla}_x G_x(x)$ is full column rank, we can choose $\epsilon_x(z)$ large enough such that the solution of (3.14) is well defined and unique. To show that, let us look at (3.14) as an optimization in its own right. Let $\bar{d}_\nu$ be the Lagrange multiplier and define the function $\bar{g}(\bar{d}_x, \bar{d}_\nu)$ which is the function $g(z,b)$ defined in (3.10) but now for problem (3.14):

$$\bar{g}(\bar{d}_x, \bar{d}_\nu) := \begin{bmatrix} \boldsymbol{\nabla}_x L(z) + (\boldsymbol{\nabla}_{xx} L(z) + \epsilon_x(z) I) \bar{d}_x + \boldsymbol{\nabla}_x G_x(x) \bar{d}_\lambda \\ G_x(x) + \boldsymbol{\nabla}_x G_x(x)' \bar{d}_x \end{bmatrix}. \tag{3.15}$$

[1] Notice that we use the second order linearization of the Lagrangian $L(z)$ as the cost function in (3.14), not the one of $f(x)$. The justification is that, if x^* is a local minimum of (3.9) with associated Lagrange multiplier ν^*, then x^* is also a local minimum of

$$\min_{x : G_x(x) = \mathbf{0}} f(x) + \nu_x^{*\prime} G_x(x).$$

Evidently, ν_x^* is not know in advance, so instead one uses the value of ν_x at the current iteration, which leads to the local approximation (3.14).

So if one takes any $\epsilon_x(z) \geqslant 0$ large enough such that

$$\operatorname{In}\left(\begin{bmatrix} \boldsymbol{\nabla}_{xx}L(z)+\epsilon_x(z) & \boldsymbol{\nabla}_x G_x(x) \\ \boldsymbol{\nabla}_x G_x(x)' & 0 \end{bmatrix}\right) = (n_x, l_x, 0), \tag{3.16}$$

then we guarantee that any point $\bar{d}_x, \bar{d}_\nu$ that satisfies $\bar{g}(\bar{d}_x, \bar{d}_\nu) = \mathbf{0}$ will be a strict local minimum of (3.14) (see Proposition 3.1.5). Moreover, this choice of $\epsilon_x(z)$ also guarantees that (3.14) is a strongly convex quadratic program, which, with the fact that $\boldsymbol{\nabla}_x G_x(x)$ is full column rank, means that the solution $(\bar{d}_x, \bar{d}_\nu)$ is *unique.* Therefore, we will take the update directions (d_x, d_ν) to be the solution $(\bar{d}_x, \bar{d}_\nu)$. Moreover, with some algebra, one can show that the solution to (3.14) is given by

$$\begin{aligned}\begin{bmatrix} d_x \\ d_\nu \end{bmatrix} &= -\begin{bmatrix} \boldsymbol{\nabla}_{xx}L(z)+\epsilon_x(z) & \boldsymbol{\nabla}_x G_x(x) \\ \boldsymbol{\nabla}_x G_x(x)' & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{\nabla}_x L(z) \\ G_x(x) \end{bmatrix} \\ &= -(\boldsymbol{\nabla}_z g(x,b)' + \operatorname{diag}([\epsilon_x(z)\mathbf{1}_{n_x}, \mathbf{0}_{l_x}]))^{-1} g(x,b).\end{aligned}$$

Let us now address the case in which there there are inequality constraints. The challenge is to take into account the constraint $s_x \geqslant 0$. To address this, let us start by relaxing the inequality constraint from (3.9) and including it in the cost as the barrier function $-b\mathbf{1}'\log(s_x)$ (the $\log(\cdot)$ is element wise).

$$\min_{x,s_x:G_x(x)=\mathbf{0},F_x(x)+s_x=\mathbf{0}} f(x) - b\mathbf{1}'\log(s_x). \tag{3.17}$$

This is a relaxation because $-b\mathbf{1}'\log(s_x)$ only accepts $s \geqslant 0$ and goes to $+\infty$ if $s_x \to 0$. The optimization (3.17) only has equality constraints, so similar to what we did in (3.14), let us construct a local second order approximation of (3.17) around z:

$$\begin{aligned}\min_{\substack{\bar{d}_x,\bar{d}_s: \\ G_x(x)+\boldsymbol{\nabla}_x G_x(x)'\bar{d}_x=\mathbf{0}, \\ F_x(x)+s_x+\boldsymbol{\nabla}_x F_x(x)'\bar{d}_x+\bar{d}_s=\mathbf{0}}} \quad & L(z) - b\mathbf{1}'\log(s_x) + \bar{d}_x'\boldsymbol{\nabla}_x L(z) + \bar{d}_s'(\lambda_x - b\mathbf{1}\oslash s_x) \\ & + \frac{1}{2}\bar{d}_x'(\boldsymbol{\nabla}_{xx}L(z)+\epsilon_x(z)I)\bar{d}_x + \frac{1}{2}\bar{d}_s'\operatorname{diag}(\lambda_x \oslash s_x)\bar{d}_s \end{aligned} \tag{3.18}$$

where $\oslash$ designates the element wise division of two vectors. Equation (3.18) is not exactly a second order approximation because instead of using as quadratic term for $\bar{d}_s$ the matrix $b\,\mathrm{diag}(s_x)^{-2}$ (which is the actual matrix given by second order approximation of $-b\mathbf{1}'\log(s_x + d_s)$ around s_x), we used the matrix $\mathrm{diag}(\lambda_x \oslash s_x)$. This is a relatively well known substitutions for interior-point methods, and is what makes it be a primal-dual interior-point method instead of a barrier interior-point method. The technical justification is that, if we were at a point such that $g(z,b) = \mathbf{0}$, the two would be equivalent as $\lambda_x \odot s_x - b\mathbf{1} = \mathbf{0}$. In practice, it has been observed that this modified linearization tends to perform better because it provides directions d_s that also take into account the current value of λ_x in the quadratic form, which helps to get a direction d_z that does no violate the constraints $\lambda_x, s_x > 0$ [54, Chapter 19.3].

Because (3.18) is a quadratic program with linear equality constraints, just as it was the case for (3.14), we can use the exact same reasoning to choose $\epsilon_x(z)$. Let us define the matrices

$$J_{zz}f(z) = \begin{bmatrix} \nabla_{xx}L(z) & 0 & \nabla_x G_x(x) & \nabla_x F_x(x) \\ 0 & \mathrm{diag}(\lambda_x \oslash s_x) & 0 & I \\ \nabla_x G_x(x)' & 0 & 0 & 0 \\ \nabla_x F_x(x)' & I & 0 & 0 \end{bmatrix} \tag{3.19}$$

and $E(z) := \mathrm{diag}(\epsilon_x(z)\mathbf{1}_{n_x}, \mathbf{0}_{m_x+l_x+m_x})$. If $\epsilon_x(z)$ is chosen large enough such that $\mathrm{In}(J_{zz} + E(z)) = (n_x + m_x, l_x + m_x, 0)$, then the solution $(\bar{d}_x, \bar{d}_s)$ of (3.18) and associated Lagrange multipliers $(\bar{d}_\nu, \bar{d}_\lambda)$ are unique. With some algebra, one could show that the solution of (3.18) is

$$\begin{aligned} d_z &= -(J_{zz}f(z) + E(z))^{-1} S^{-1} g(z,b) \\ &= -(\nabla_z g(z,b)' + E(z))^{-1} g(z,b) \end{aligned}$$

where $S := \mathrm{diag}(\mathbf{1}_{n_x}, s_x, \mathbf{1}_{l_x+m_x})$. Putting it all together, the modified primal-dual interior-point is governed by the equation

$$z^+ = z + \alpha d_z = z - \alpha(\boldsymbol{\nabla}_z g(z,b)' + E(z))^{-1} g(z,b), \tag{3.20}$$

where $\alpha \in (0,1]$ is chosen such that $\lambda_x, s_x > \mathbf{0}$. Conveniently, because we used $\operatorname{diag}(\lambda_x \oslash s_x)$ for the second order linearization of the barrier, when $\epsilon_x(x) = 0$, we recover the basic primal-dual interior-point method from (3.13). We refer to [54, Chapter 19] for a complete description of an algorithm using (3.20), including a strategy to decrease the barrier parameter b. Alternatively, we describe such strategy in Section 3.3 for the minmax optimization case.

We can now state a result connecting the stability/instability of any equilibrium point of the modified primal-dual interior-point method to such point being or not a local minimum. The theorem says essentially the same thing as Theorem 3.1.3: On the one hand, even if $\operatorname{In}(J_{zz}f(z)) = (n_x + m_x, l_x + m_x, 0)$, taking $\epsilon_x(z) > 0$ will not impair the algorithm's capacity to converge towards a local minimum; this can be useful, for instance, if $\operatorname{In}(J_{zz}f(z))$ has an eigenvalue close to 0. On the other hand, using the modified primal-dual interior-point method essentially guarantees that the algorithm can only converge towards and equilibrium point if such point is a local minimum, thus fixing the issue of primal-dual interior-point methods being attracted to any equilibrium point, regardless of whether such point is a local minimum.

Theorem 3.1.6 (Stability of modified interior-point method for minimization) *Let $\alpha = 1$ and (z,b) with $b > 0$, be an equilibrium point in the sense that $g(z,b) = \mathbf{0}$. Assume the LICQ and strict complementarity hold at z, that $J_{zz}f(z)$ is invertible, and that $J_{zz}f(\cdot)$ is differentiable on a neighborhood around z. Then for any function $\epsilon_x(\cdot)$ that is constant in a neighborhood around z and satisfies $\operatorname{In}(J_{zz} + E(z)) = (n_x + m_x, l_x + m_x, 0)$ one has that if:*

i) z is a local minimum, then it is a LAS equilibrium point of (3.20)*.*

ii) z is not a local minimum, then it is an unstable equilibrium point of (3.20).

Proof sketch. First, using the same arguments as in the proof of Theorem 3.1.3, we conclude that the Jacobian of the dynamic system (3.20) around a point z for which $g(z,b)=\mathbf{0}$ is

$$I-\alpha\Big(J_{zz}f(z)+E(z)\Big)^{-1}S^{-1}\nabla_z g(z,b)' = I-\alpha\Big(J_{zz}f(z)+E(z)\Big)^{-1}J_{zz}f(z) \qquad (3.21)$$

Second, it is straightforward to check that $H_{zz}f(z)=S^{1/2}J_{zz}f(z)S^{1/2}$ which, using Sylvester's law of inertia [116, Theorem 1.5], means that $\mathrm{In}(H_{zz}f(z))=\mathrm{In}(J_{zz}f(z))$. This means that one can check the second order conditions in (3.12) by using $J_{zz}f(z)$.

Let us define the matrix

$$R(\mu)=Z_x(z)'\begin{bmatrix}\nabla_{xx}L(z)+\mu\epsilon_x(z)I & 0\\ 0 & \mathrm{diag}(\lambda_x\oslash s_x)\end{bmatrix}Z_x(z)$$

where $Z_x(z)\in\mathbb{R}^{n_x+m_x,n_x-l_x}$ is a matrix with full column rank such that

$$\begin{bmatrix}\nabla_x G_x(x)' & \mathbf{0}\\ \nabla_x F_x(x)' & I\end{bmatrix}Z_x(z)=\mathbf{0}. \qquad (3.22)$$

Using the same arguments as in the proof of Proposition 3.1.5, we conclude that

$$\mathrm{In}(J_{zz}f(z)+E(z))=\mathrm{In}(R(\mu))+(l_x+m_x,l_x+m_x),$$

which implies that $\mathrm{In}(J_{zz}f(z)+E(z))=(n_x+m_x,l_x+m_x)$ is equivalent to $R(1)\succ 0$ and that the second order sufficient condition is equivalent to $R(0)\succ 0$. This means that the rest of the theorem's proof is analogous to the one of Theorem 3.1.3, but instead of looking at the sign of the smallest eigenvalue of $\nabla_{xx}f(x)+\mu\epsilon_x(z)I$, one looks at the sign of the smallest eigenvalue of the matrix $R(\mu)$.

If z is a local minimum, then $\lambda_{min}(R(0))>0$. As $\epsilon_x(z)\geqslant 0$, we conclude that $\lambda_{min}(R(\mu))>0$ for every $\mu\geqslant 0$ and therefore z is a LAS equilibrium point of (3.13).

Conversely, if z is not a local minimum, $\lambda_{min}(R(0)) < 0$. By construction, $\epsilon_x(z)$ is such that $\lambda_{min}(R(1)) > 0$, therefore, by continuity of the eigenvalue, there is a $\mu \in (0,1)$ such that $\lambda_{min}(R(\mu)) = 0$ and therefore z is an unstable equilibrium point of (3.13).

3.2 Minmax optimization

Consider the minmax optimization problem

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}(x)} f(x,y) \tag{3.23}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is a twice continuously differentiable objective function, $\mathcal{X} \subset \mathbb{R}^{n_x}$ is the feasible set for x and $\mathcal{Y} : \mathcal{X} \rightrightarrows \mathbb{R}^{n_y}$ is a set-valued map that defines an x dependent feasible set for y; we do not make any convexity or concavity assumption on $f(\cdot)$, $\mathcal{X}$ and $\mathcal{Y}(\cdot)$. We chose $\mathcal{Y}(\cdot)$ to be dependent on x because this describes the most general application. Moreover, having the constraints of the inner maximization to depend on the value of outer maximization is often necessary in problems such as robust Model Predictive Control or in bi-level optimization. Furthermore, notice that we do not make any assumption on whether the min and the max commute (and this would not be well defined as $\mathcal{Y}(\cdot)$ depends on x). A solution (x^*, y^*) to (3.23) is called a global minmax and satisfies

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant \max_{\tilde{y} \in \mathcal{Y}(x)} f(x, \tilde{y}) \qquad \forall (x,y) \in \mathcal{X} \times \mathcal{Y}(x^*).$$

A point (x^*, y^*) is said to be a local minmax of (3.23) if there exist a constant $\delta_0 > 0$ and a positive function $h(\cdot)$ satisfying $h(\delta) \to 0$ as $\delta \to 0$, such that for every $\delta \in (0, \delta_0]$ and for every $(x,y) \in \{x \in \mathcal{X} : \|x - x^*\| \leqslant \delta\} \times \{y \in \mathcal{Y}(x^*) : \|y - y^*\| \leqslant h(\delta)\}$ we have

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant \max_{\tilde{y} \in \mathcal{Y}(x) : \|\tilde{y} - y^*\| \leqslant h(\delta)} f(x, \tilde{y})$$

[52,53]. Inspired by the properties of the modified Newton and primal-dual interior-point methods for minimization in Section 3.1, we want to develop a Newton-type iterative algorithm of the form

$$\begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}. \tag{3.24}$$

where d_x and d_y satisfy the following properties:

P1: At each time step, (d_x, d_y) is obtained from the solution of a quadratic program that locally approximates (3.23) and therefore (x^+, y^+) can be seen as an improvement over (x, y). This acts as a surrogate for guiding the modified Newton's method towards a solution at each step.

P2: The iterations of (3.24) can converge towards an equilibrium point only if such point is a local minmax. Similar to what was the case in minimization (see Example 3.1.1), a pure Newton method will be attracted to any equilibrium point. This makes sure that the iterations will not be attracted to equilibrium points that are not local minmax.

P3: The iterations of (3.24) can converge to any local minmax. This property means that any modification to Newton's method needs to keep local minmax as attractor.

3.2.1 Unconstrained minmax

We start by considering the case where $\mathcal{X} = \mathbb{R}^{n_x}$ and $\mathcal{Y}(\cdot) = \mathbb{R}^{n_y}$ such that (3.23) simplifies to

$$\min_{x \in \mathbb{R}^{n_x}} \max_{y \in \mathbb{R}^{n_y}} f(x, y). \tag{3.25}$$

For this case, [52] establishes second order sufficient conditions to determine if a point (x, y) is a local minmax which can be stated in terms of the inertia of the matrix

$$\boldsymbol{\nabla}_{zz} f(x,y) := \begin{bmatrix} \boldsymbol{\nabla}_{xx} f(x,y) & \boldsymbol{\nabla}_{xy} f(x,y) \\ \boldsymbol{\nabla}_{yx} f(x,y) & \boldsymbol{\nabla}_{yy} f(x,y) \end{bmatrix}.$$

We recall that the inertia $\mathrm{In}(A)$ of a symmetric matrix A is a 3-tuple with the number of positive, negative and zero eigenvalues of A.

Proposition 3.2.1 (Second order conditions for unconstrained minmax) *Let (x, y) be an equilibrium point in the sense that $\boldsymbol{\nabla}_x f(x, y) = 0$ and $\boldsymbol{\nabla}_y f(x, y) = 0$. If*

$$\mathrm{In}(\boldsymbol{\nabla}_{yy} f(x,y)) = (0, n_y, 0) \text{ and } \mathrm{In}(\boldsymbol{\nabla}_{zz} f(x,y)) = (n_x, n_y, 0) \tag{3.26}$$

then (x, y) is a local minmax.

The second order conditions in [52] are:

$$\mathrm{In}(\boldsymbol{\nabla}_{yy} f(x,y)) = (0, n_y, 0) \text{ and}$$
$$\mathrm{In}(\boldsymbol{\nabla}_{xx} f(x,y) - \boldsymbol{\nabla}_{xy} f(x,y) \boldsymbol{\nabla}_{yy} f(x,y)^{-1} \boldsymbol{\nabla}_{yx} f(x,y)) = (n_x, 0, 0),$$

which turn out to be equivalent to the inertia conditions in Proposition 3.2.1 in view of Haynsworth inertia additivity formula [116, Theorem 1.6]. Notice that the second order sufficient conditions are *not* symmetric. A point might be a local minmax even if $\boldsymbol{\nabla}_{xx} f(x,y) \nsucc 0$ as long as $-\boldsymbol{\nabla}_{xy} f(x,y) \boldsymbol{\nabla}_{yy} f(x,y)^{-1} \boldsymbol{\nabla}_{yx} f(x,y)$ (which is positive) is large enough. So the second order conditions are what allow one to distinguish between an equilibrium point being a local minmax and a minmin, maxmax or maxmin. One can interpret the second order sufficient conditions as saying that $y \mapsto f(x, y)$ is strongly concave in a neighborhood around (x, y) and $x \mapsto \max_{\tilde{y}:\|y-\tilde{y}\|<\delta} f(x, \tilde{y})$ as being strongly convex in a neighborhood around (x, y) for some $\delta > 0$. Notice that these are only local properties around local minmax, as $f(\cdot)$ may be nonconvex-nonconcave away from local minmax points.

In order to obtain property P1, we propose to obtain the Newton direction (d_x, d_y) for (3.24) by solving the following local quadratic approximation to (3.25)

$$\begin{aligned}\min_{\bar{d}_x}\max_{\bar{d}_y} & f(x,y) + \boldsymbol{\nabla}_x f(x,y)'\bar{d}_x + \boldsymbol{\nabla}_y f(x,y)'\bar{d}_y + \bar{d}_x'\boldsymbol{\nabla}_{xy} f(x,y)\bar{d}_y \\ &+ \frac{1}{2}\bar{d}_x'\Big(\boldsymbol{\nabla}_{xx} f(x,y) + \epsilon_x(x,y)I\Big)\bar{d}_x + \frac{1}{2}\bar{d}_y'\Big(\boldsymbol{\nabla}_{yy} f(x,y) - \epsilon_y(x,y)I\Big)\bar{d}_y \end{aligned} \tag{3.27}$$

with $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ chosen so that the minmax problem in (3.27) has a unique solution, which means that the inner (quadratic) maximization must be strictly concave and that the outer (quadratic) minimization of the maximized function must be strictly convex, which turns out to be precisely the second order sufficient conditions in Proposition 3.2.1, applied to the approximation in (3.27), which can be explicitly written as follows:

$$\begin{aligned}\text{In}\Big(\boldsymbol{\nabla}_{yy} f(x,y) - \epsilon_y(x,y)I\Big) &= (0, n_y, 0) \text{ and} \\ \text{In}\Big(\boldsymbol{\nabla}_{zz} f(x,y) + E(x,y)\Big) &= (n_x, n_y, 0)\end{aligned} \tag{LQAC}$$

where $E(x,y) = \text{diag}(\epsilon_x(x,y)\mathbf{1}_{n_x}, -\epsilon_y(x,y)\mathbf{1}_{n_y})$. We call these condition the Local Quadratic Approximation Condition (LQAC). It is straightforward to show that the Newton iterations (3.24) with (d_x, d_y) obtained from the solution to (3.27) is given by

$$\begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \Big(\boldsymbol{\nabla}_{zz} f(x,y) + E(x,y)\Big)^{-1} \begin{bmatrix} \boldsymbol{\nabla}_x f(x,y) \\ \boldsymbol{\nabla}_y f(x,y) \end{bmatrix}. \tag{3.28}$$

To obtain properties P2 and P3, we need all locally asymptotically stable equilibrium points of (3.27) to be local minmax of (3.25) and that all other equilibrium points of (3.27) to be unstable. For the unconstrained minimization in Section 3.1.1, to obtain the equivalent of properties P2 and P3 it was sufficient to simply select $\epsilon_x(\cdot)$ such that the local quadratic approximation (3.4) has a well-defined minimum (Theorem 3.1.3).

However, for minmax optimization the (LQAC) does not suffice to guarantee that P2 and P3 hold. Our first counter example bellow show how the (LQAC) are not enough to ensure that P2 holds; our second counter example show how they are not enough to guarantee that P3 holds.

Example 3.2.2 *Consider* $f(x,y) = 1.5x^2 - 4xy + y^2$ *for which the unique equilibrium point* $x = y = 0$ *is not a local minmax point. Take* $\epsilon_y(0,0) = 4$ *and* $\epsilon_x(0,0) = 0$ *which satisfy* (LQAC)*. The Jacobian of the dynamics is*

$$I - \left(\begin{bmatrix} 3 & -4 \\ -4 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -4 \end{bmatrix} \right)^{-1} \begin{bmatrix} 3 & -4 \\ -4 & 2 \end{bmatrix} \approx \begin{bmatrix} 0 & 0.72 \\ 0 & 0.54 \end{bmatrix}$$

which has eigenvalues approximately equal to $(0, 0.54)$*. Therefore* $(0,0)$ *is a LAS equilibrium point of* (3.28) *even though it is not a local minmax point.*

Example 3.2.3 *Consider* $f(x,y) := -0.25x^2 + xy - 0.5y^2$*, for which the unique equilibrium point* $x = y = 0$ *is a local minmax point. Take* $\epsilon_y(0,0) = 3$ *and* $\epsilon_x(0,0) = 0.2$ *which satisfy* (LQAC)*. The Jacobian of the dynamics is*

$$I - \left(\begin{bmatrix} -0.5 & 1 \\ 1 & -1 \end{bmatrix} + \begin{bmatrix} 0.3 & 0 \\ 0 & -3 \end{bmatrix} \right)^{-1} \begin{bmatrix} -0.5 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 6 & -15 \\ 1.5 & -3 \end{bmatrix},$$

for which the eigenvalues are $1.5 \pm 1.5i$*. Therefore* $(0,0)$ *is an unstable equilibrium point of* (3.28) *even though it is a local minmax point.*

The main contribution of this section is a set of sufficient conditions that, in addition to (LQAC), guarantee P2 and P3 hold.

Theorem 3.2.4 (Stability of modified Newton's method for minmax) *Let* (x,y) *be an equilibrium point in the sense that* $\nabla_x f(x,y) = \mathbf{0}$ *and* $\nabla_y f(x,y) = \mathbf{0}$*. Assume that*

$\nabla_{zz} f(x, y)$ and $\nabla_{yy} f(x, y)$ are invertible and that $\nabla_{zz} f(\cdot)$ is differentiable on a neighborhood around (x, y). Then there exist functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ that are constant in a neighborhood around (x, y), satisfy the (LQAC) *at (x, y) and guarantee that if:*

i) (x, y) is a local minmax, then it is a LAS equilibrium point of (3.28).

ii) (x, y) is not a local minmax, then it is an unstable equilibrium point of (3.28).

The theorem's implications are similar to those of Theorem 3.1.3. On the one hand, if (x, y) is a local minmax, then it is possible to construct functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ that guarantee that the modified Newton method can converge towards a local minmax. A natural choice for such function near a local minmax is to take $\epsilon_y(\cdot) = \epsilon_x(\cdot) = 0$, which not only provides the stability result, but can also achieve superlinear convergence if $f(\cdot)$ is Lipschitz. On the other hand, if (x, y) is an equilibrium point but not a local minmax, it is possible to construct functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ such that the algorithm's iterations *cannot* converge towards it. This means that the modified Newton's method for minmax can only converge towards an equilibrium point if such point is a local minmax.

While the statement of Theorem 3.2.4 is about existence, the proof is actually constructive. The functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ are not unique, and have to satisfy the following conditions:

i) For the stability result, if $\epsilon_y(x, y) = 0$, then the stability property is guaranteed by any $\epsilon_x(x, y) \geqslant 0$. If $\epsilon_y(x, y) > 0$, then $\epsilon_x(x, y)$ needs to be taken large enough to satisfy the condition in equation (3.31) of the proof.

ii) For the instability result:

- unless $\text{In}(\nabla_{yy} f(x, y)) \neq (0, n_y, 0)$ and $\text{In}(\nabla_{zz} f(x, y)) = (n_x, n_y, 0)$, then it is sufficient for $\epsilon_x(x, y)$ and $\epsilon_y(x, y)$ to satisfy the (LQAC) to guarantee instability.

- if $\mathrm{In}(\boldsymbol{\nabla}_{yy}f(x,y)) \neq (0,n_y,0)$ and $\mathrm{In}(\boldsymbol{\nabla}_{zz}f(x,y)) = (n_x,n_y,0)$ then for a given $\epsilon_y(x,y)$, $\epsilon_x(x,y)$ needs to be large enough such that for some $\mu \in (0,1)$, $\mathrm{In}(\boldsymbol{\nabla}_{zz}f(x,y) + \mu E(x,y)) \neq (n_x,n_y,0)$.

We use these results in Section 3.3 to present an efficient way to numerically construct these functions.

Proof of Theorem 3.2.4. The fact that the (LQAC) can always be satisfied is straightforward: as $\boldsymbol{\nabla}_{zz}f(x,y)$ is differentiable, its eigenvalues are bounded and can be made have the desired inertia by taking sufficiently large (but finite) values of $\epsilon_x(x,y)$ and $\epsilon_y(x,y)$. Moreover, from our assumption that $\boldsymbol{\nabla}_{zz}f(x,y)$ and $\boldsymbol{\nabla}_{yy}f(x,y)$ are invertible, (x,y) is a local minmax point if and only if (x,y) satisfy the second order sufficient in (3.26); this is implied by the second order necessary conditions for local minmax in [52].

Using the same reasoning as in Theorem 3.1.3, as the (LQAC) hold then $(\boldsymbol{\nabla}_{zz}f(x,y) + E(x,y))$ is nonsingular and the Jacobian of the dynamical system (3.28) at (x,y) is

$$I - (\boldsymbol{\nabla}_{zz}f(x,y) + E(x,y))^{-1}\boldsymbol{\nabla}_{zz}f(x,y). \tag{3.29}$$

Therefore, we can also use the same reasoning as in the proof of Theorem 3.1.3 to conclude that (x,y) is a LAS equilibrium point of (3.28) if $\boldsymbol{\nabla}_{zz}f(x,y)+\mu E(x,y)$ is nonsingular $\forall \mu \in [0,2]$. Conversely, (x,y) is an unstable equilibrium point of (3.28) if $\boldsymbol{\nabla}_{zz}f(x,y)+\mu E(x,y)$ is singular for some $\mu \in (0,2)$.

For the rest of the proof, it will be useful to have defined the function

$$R(\mu) = \boldsymbol{\nabla}_{xx}f(x,y) - \boldsymbol{\nabla}_{xy}f(x,y)(\boldsymbol{\nabla}_{yy}f(x,y) - \mu\epsilon_y(x,y)I)^{-1}\boldsymbol{\nabla}_{yx}f(x,y) + \mu\epsilon_x I \tag{3.30}$$

and to drop the inputs (x,y) from the expressions in order to shorten them.

Let us start by proving the statement for the case when (x,y) is a local minmax, in which case the (LQAC) hold with $\epsilon_y = \epsilon_x = 0$. We will prove that if

$$\epsilon_x \geqslant \lambda_{min}(\epsilon_y \boldsymbol{\nabla}_{xy}f \boldsymbol{\nabla}_{yy}f^{-2}\boldsymbol{\nabla}_{yx}f). \tag{3.31}$$

then (x, y) is a LAS equilibrium point of (3.28). To prove it, we will show (3.31) ensures that $\nabla_{zz} f + \mu E$ is nonsingular $\forall\ \mu \geqslant 0$. First, as $\nabla_{yy} f \prec 0$, $\mu \geqslant 0$, and $\epsilon_y \geqslant 0$, we have $\nabla_{yy} f - \mu\epsilon_y I \prec 0$ and is thus nonsingular. Second, let us show that the condition (3.31) implies that for any vector v

$$\min_{\mu \in [0,2]} v' R(\mu) v = v' R(0) v. \tag{3.32}$$

Taking the derivative of $v'R(\mu)v$ with respect to μ we obtain

$$v'\Big(\epsilon_x I - \epsilon_y \nabla_{xy} f (\nabla_{yy} f - \mu\epsilon_y I)^{-2} \nabla_{yx} f\Big) v \succ v'\Big(\epsilon_x I - \epsilon_y \nabla_{xy} f \nabla_{yy} f^{-2} \nabla_{yx} f\Big) v$$

in which we use the the fact that $\nabla_{yy} f^{-2} \succeq (\nabla_{yy} f - \mu\epsilon_y I)^{-2}$ for all $\mu \geqslant 0$ as $\nabla_{yy} f \prec 0$, and $\epsilon_y \geqslant 0$. Therefore, if (3.31) holds, the derivative of $v'R(\mu)v$ with respect to μ is non-negative, thus the cost does not decrease with μ, which implies that the minimum is obtained for $\mu = 0$, which proves (3.32). Therefore if ϵ_x and ϵ_y are chosen to satisfy (3.31), then $\forall \mu \in [0, 2]$ it holds that $R(\mu) \succeq R(0) \succ 0I$, where the second inequality comes from the second order sufficient conditions for unconstrained minmax (3.26). As neither $\nabla_{yy} f - \mu\epsilon_y I \prec 0$ nor $R(\mu)$ are singular for $\mu \in [0, 2]$, Haynsworth inertia additivity formula [116, Theorem 1.6] implies that $\nabla_{zz} f + \mu E$ is nonsingular $\forall \mu \in [0, 2]$, and therefore (x, y) is a LAS equilibrium point of (3.28).

Now the second part, let us prove the statement for the case in which (x, y) is not a local minmax. We will show that for every ϵ_y such that $\in (\nabla_{yy} f - \epsilon_y I) = (0, n_y, 0)$ there for any large enough ϵ_x, the (LQAC) are satisfied and

$$\nabla_{zz} f + \mu \operatorname{diag}(\epsilon_x \mathbf{1}_{n_x}, -\epsilon_y \mathbf{1}_{n_y}) = \nabla_{zz} f + \mu E \tag{3.33}$$

is singular for some $\mu \in (0, 1)$, which in turn guarantees that (x, y) is an unstable equilibrium point of (3.28) (see discussion in the beginning of the proof).

If $\mathrm{In}(\boldsymbol{\nabla}_{zz}f) \neq (n_x, n_y, 0)$, then any large enough value of ϵ_x such that (LQAC) holds is enough to guarantee that $\boldsymbol{\nabla}_{zz}f + \mu E$ is singular for some $\mu \in (0,1)$. The proof is straightforward: If $\mathrm{In}(\boldsymbol{\nabla}_{zz}f) \neq (n_x, n_y, 0)$ and $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + E) = (n_x, n_y, 0)$ (from the (LQAC)), then, by continuity of the eigenvalue $\exists \mu \in (0,1)$ such that $\boldsymbol{\nabla}_{zz}f + \mu E$ is singular.

If $\mathrm{In}(\boldsymbol{\nabla}_{zz}f) = (n_x, n_y, 0)$ but $\mathrm{In}(\boldsymbol{\nabla}_{yy}f) \neq (0, n_y, 0)$, then the (LQAC) is not enough to guarantee that (x, y) is an unstable equilibrium point. However, it is possible to guarantee instability. The proof is the following.

Let μ^* be the largest $\mu \in (0,1)$ such that $\boldsymbol{\nabla}_{yy}f - \mu\epsilon_y I$ is singular. We know that this point exists because, on the one hand, by assumption $\boldsymbol{\nabla}_{yy}f$ is invertible (and therefore $\mu^* > 0$), and on the other hand, we know that $\boldsymbol{\nabla}_{yy}f \nprec 0$ and that $\boldsymbol{\nabla}_{yy}f - \epsilon_y I \prec 0$ by construction (and therefore $\mu^* < 1$).

Now take any $\bar{\mu} \in (0, \mu^*)$ such that $\boldsymbol{\nabla}_{yy}f - \bar{\mu}\epsilon_y I$ is invertible (there are uncountable many). Suppose there exists $\bar{\epsilon}$ such that for any $\epsilon_x \geqslant \bar{\epsilon}$, the (LQAC) hold and $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \bar{\mu}E) \neq (n_x, n_y, 0)$. If such $\bar{\epsilon}$ exists, then, by the continuity of the eigenvalues, if $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \bar{\mu}E) \neq (n_x, n_y, 0)$ this means that $\boldsymbol{\nabla}_{zz}f + \mu E$ is singular for some $\mu \in (0, \bar{\mu}]$.

So, to conclude the proof, we just need to show the existence of such $\bar{\epsilon}$. Take any ϵ_x such that $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \bar{\mu}E) = (n_x, n_y, 0)$ (otherwise the proof is tautological). From Haynsworth inertia additivity formula, we have that

$$\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \bar{\mu}E) = \mathrm{In}(R(\bar{\mu})) + \mathrm{In}(\boldsymbol{\nabla}_{yy}f - \bar{\mu}\epsilon_y I)$$

with $\mathrm{In}(R(\bar{\mu})) = (n_x - k, k, 0)$ and $\mathrm{In}(\boldsymbol{\nabla}_{yy}f - \bar{\mu}\epsilon_y I) = (k, n_y - k, 0)$ for some $k \in \{1, \dots, \min(n_x, n_y)\}$. On the one hand, it is straightforward to establish that $\exists \bar{\epsilon}_1$ such that if $\epsilon_x \geqslant \bar{\epsilon}_1$, then $\mathrm{In}(R(\bar{\mu})) \neq (n_x - k, k, 0)$, which means that $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \bar{\mu}E) \neq (n_x, n_y, 0)$. On the other hand, $\exists \bar{\epsilon}_2$ such that if $\epsilon_x \geqslant \bar{\epsilon}_2$, then $\mathrm{In}(\boldsymbol{\nabla}_{zz}f + \mu E) = (n_x, n_y, 0)$. Therefore, we can define $\bar{\epsilon} = \max(\bar{\epsilon}_1, \bar{\epsilon}_2)$, which concludes the proof

3.2.2 Constrained minmax

We now consider the case with more general constraint sets involving equality and inequality constraints of the form

$$\begin{aligned} \mathcal{X} &= \{x \in \mathbb{R}^{n_x} : G_x(x) = \mathbf{0}, F_x(x) \leqslant \mathbf{0}\} \quad \text{and} \\ \mathcal{Y}(x) &= \{y \in \mathbb{R}^{n_y} : G_y(x,y) = \mathbf{0}, F_y(x,y) \leqslant \mathbf{0}\} \end{aligned} \tag{3.34}$$

where the functions $G_x : \mathbb{R}^{n_x} \to \mathbb{R}^{l_x}$, $F_x : \mathbb{R}^{n_x} \to \mathbb{R}^{m_x}$, $G_y : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{l_y}$ and $F_y : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^{m_y}$ are all twice continuously differentiable. Similar to what we did in Section 3.1.2, it will be convenient for the development of the primal-dual interior-point method to use slack variables and rewrite the constrained minmax (3.23) as

$$\min_{x,s_x : G_x(x)=\mathbf{0}, F_x(x)+s_x=\mathbf{0}, s_x \geqslant \mathbf{0}} \quad \max_{y,s_y : G_y(x,y)=\mathbf{0}, F_y(x,y)+s_y=\mathbf{0}, s_y \geqslant \mathbf{0}} f(x,y). \tag{3.35}$$

where $s_x \in \mathbb{R}^{m_x}$ and $s_y \in \mathbb{R}^{m_y}$.

Similar to what we have done in the unconstrained case, we want to present second order conditions to determine if a point is a constrained local minmax. In order to do so, we need to extend some fundamental concepts of constrained minimization to constrained minmax optimization. The function

$$L(z) := f(x,y) + \nu_x' G_x(x) + \lambda_x'(F_x(x) + s_x) + \nu_y' G_y(x,y) - \lambda_y'(F_y(x,y) + s_y),$$

will play an equivalent role as the Lagrangian with $(\nu_x, \nu_y, \lambda_x, \lambda_y)$ as the equivalent of Lagrange multipliers; we use the shorthand notation $z = (x, s_x, y, s_y, \nu_y, \lambda_y, \nu_x, \lambda_x)$. Furthermore, we use the following definition of linear independence constraint qualifications (LICQ) and of strict complementarity for minmax optimization:

Definition 3.2.5 (LICQ and strict complementarity for minmax) *Let the sets of active inequality constraints for the minimization and maximization be defined, respec-*

tively, by

$$\mathcal{A}_x(x) = \{i : F_x^{(i)}(x) = 0, i = 1, \ldots, m_x\} \textit{ and}$$
$$\mathcal{A}_y(x, y) = \{i : F_y^{(i)}(x, y) = 0, i = 1, \ldots, m_y\}$$

where $F_x^{(i)}(x)$ and $F_y^{(i)}(x, y)$ denote the i^{th} element of $F_x(x)$ and $F_y(x, y)$. Then:

- *The linear independence constraint qualification (LICQ) is said to hold at z if the vectors in the sets*

  $$\{\boldsymbol{\nabla}_x G_x^{(i)}(x), i = 1, \ldots, l_x\} \bigcup \{\boldsymbol{\nabla}_x F_x^{(i)}(x), i \in \mathcal{A}_x(x)\} \textit{ and}$$
  $$\{\boldsymbol{\nabla}_y G_y^{(i)}(x, y), i = 1, \ldots, l_y\} \bigcup \{\boldsymbol{\nabla}_y F_y^{(i)}(x, y), i \in \mathcal{A}_y(x, y)\}$$

  are linearly independent.

- *Strict complementarity is said to hold at z if $\lambda_y^{(i)} > 0 \ \forall i \in \mathcal{A}_y(x, y)$ and $\lambda_x^{(i)} > 0 \ \forall i \in \mathcal{A}_x(x)$*

We have almost all the ingredients to present the second order conditions for constrained minimization. For the unconstrained minmax optimization, the second order condition in Proposition 3.2.1 required that gradients ($\boldsymbol{\nabla}_x f(x, y)$ and $\boldsymbol{\nabla}_y f(x, y)$) were equal to zero and that Hessians ($\boldsymbol{\nabla}_{zz} f(x, y)$ and $\boldsymbol{\nabla}_{yy} f(x, y)$) had a particular inertia. Analogously to what was the case for the constrained minimization in Section 3.1.2, if it were not for the inequality constraints in (3.34), we would be able to state the second order conditions using gradients and Hessians of $L(z)$. The inequality constraints make the statement a bit more complicated. The role of the gradient will be played by

$$g(z,b) := \begin{bmatrix} \boldsymbol{\nabla}_x L(z) \\ \lambda_x \odot s_x - b\mathbf{1} \\ \boldsymbol{\nabla}_y L(z) \\ -\lambda_y \odot s_y + b\mathbf{1} \\ G_y(x,y) \\ -F_y(x,y) - s_y \\ G_x(x) \\ F_x(x) + s_x \end{bmatrix}$$

where $\odot$ denotes the element wise Hadamard product of two vectors and $b \geqslant 0$ the barrier parameter, which is the extension to minmax of the function $g(\cdot)$ defined in (3.10) for the minimization. The role of $\boldsymbol{\nabla}_{yy} f(x,y)$ will be played by

$$H_{yy} f(z) = \begin{bmatrix} \boldsymbol{\nabla}_{yy} L(z) & \mathbf{0} & \boldsymbol{\nabla}_y G_y(x,y) & -\boldsymbol{\nabla}_y F_y(x,y) \\ \mathbf{0} & -\operatorname{diag}(\lambda_y) & \mathbf{0} & -\operatorname{diag}(s_y^{1/2}) \\ \boldsymbol{\nabla}_y G_y(x,y)' & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{\nabla}_y F_y(x,y)' & -\operatorname{diag}(s_y^{1/2}) & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{3.36a}$$

while the role of $\boldsymbol{\nabla}_{zz} f(x,y)$ will be played by

$$H_{zz} f(z) = \begin{bmatrix} H_{xx} f(z) & H_{xy} f(z) & H_{x\lambda} f(z) \\ H_{xy} f(z)' & H_{yy} f(z) & \mathbf{0} \\ H_{x\lambda} f(z)' & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.36b}$$

with blocks defined by

$$\begin{aligned} H_{xy} f(z) &= \begin{bmatrix} \boldsymbol{\nabla}_{xy} L(z) & \mathbf{0} & \boldsymbol{\nabla}_x G_y(x,y) & -\boldsymbol{\nabla}_x F_y(x,y) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ H_{xx} f(z) &= \begin{bmatrix} \boldsymbol{\nabla}_{xx} L(z) & \mathbf{0} \\ \mathbf{0} & \operatorname{diag}(\lambda_x) \end{bmatrix} \quad H_{x\lambda} f(z) = \begin{bmatrix} \boldsymbol{\nabla}_x G_x(x) & \boldsymbol{\nabla}_x F_x(x) \\ \mathbf{0} & \operatorname{diag}(s_x^{1/2}) \end{bmatrix} \end{aligned} \tag{3.36c}$$

Proposition 3.2.6 (Second order conditions for constrained minmax) *Let z be an equilibrium point in the sense that $g(z,0)=\mathbf{0}$ with $\lambda_y, \lambda_x, s_y, s_x \geqslant \mathbf{0}$. If the LICQ and strict complementarity hold at z and*

$$\begin{aligned}\operatorname{In}(H_{yy}f(z)) &= (l_y{+}m_y, n_y+m_y, 0) \text{ and}\\ \operatorname{In}(H_{zz}f(z)) &= (n_x+m_x+l_y{+}m_y, l_x+m_x+n_y+m_y, 0)\end{aligned} \tag{3.37}$$

then (x,y) is a local minmax of (3.23).

Similar to what was the case for the second order sufficient conditions for unconstrained minmax in Proposition 3.2.1, the conditions in (3.37) are not symmetric, highlighting that there is a distinction between the minimizer and maximizer. Moreover, similar to the unconstrained case, one can interpret the second order sufficient conditions as saying that $y \mapsto f(x,y)$ is strongly concave in a neighborhood around (x,y) and $x \mapsto \max_{\tilde{y}\in\mathcal{Y}(x):\|y-\tilde{y}\|<\delta} f(x,\tilde{y})$ as being strongly convex in a neighborhood around (x,y) for some $\delta > 0$.

The conditions for Proposition 3.2.6 are slightly stricter than the ones in [53] as we require strict complementarity and LICQ both for the max and the min. However, our conditions allow us to verify whether a point is a local minmax using the inertia, instead of having to compute solution cones. We prove that given these stricter assumptions our conditions are equivalent to those in [53] in Appendix 3.A.

Primal-dual interior-point method

Let $d_z = (d_x, d_{s_x}, d_y, d_{s_y}, d_{\nu_y}, d_{\lambda_y}, d_{\nu_x}, d_{\lambda_x})$ be a shorthand notation to designate the update direction of the variables $z = (x, s_x, y, s_y, \nu_y, \lambda_y, \nu_x, \lambda_x)$. Similar to the basic primal-dual interior-point method introduced in Section 3.1.2, a basic primal-dual interior-point method for minmax finds a candidate solution to (3.35) using the iterations

$$z^+ = z + \alpha d_z = z - \alpha \boldsymbol{\nabla}_z g(z,b)'^{\,-1} g(z,b) \tag{3.38}$$

where the barrier parameter b is slowly decreased to 0, so that z converges to a root of $g(z,0) = \mathbf{0}$ while $\alpha \in (0,1]$ is chosen at each step such that the feasibility conditions $\lambda_y, \lambda_x, s_y, s_x > 0$ hold. We want to modify this basic primal-dual interior-point so it satisfies the properties P1, P2 and P3.

In order to obtain property P1, we propose to obtain d_z from the solution of a quadratic program that locally approximates (3.35). Using equivalent arguments as in the development of the quadratic program (3.18) for the constrained minimization in Section 3.1.2, we obtain that the objective function should be

$$\begin{aligned} K(d_x, d_{s_x}, d_y, d_{s_y}) = L(z) &+ \boldsymbol{\nabla}_x L(z)' d_x + (\lambda_x - b\mathbf{1} \oslash s_x)' d_{s_x} + \boldsymbol{\nabla}_y L(z)' d_y \\ &- (\lambda_y - b\mathbf{1} \oslash s_y)' d_{s_y} + d_x' \boldsymbol{\nabla}_{xy} L(z) d_y + \frac{1}{2} d_x' (\boldsymbol{\nabla}_{xx} L(z) + \epsilon_x(z) I) d_x \\ &+ \frac{1}{2} d_{s_x}' \operatorname{diag}(\lambda_x \oslash s_x) d_{s_x} + \frac{1}{2} d_y' (\boldsymbol{\nabla}_{yy} L(z) - \epsilon_y(z) I) d_y - \frac{1}{2} d_{s_y}' \operatorname{diag}(\lambda_y \oslash s_y) d_{s_y}, \end{aligned}$$

where $\epsilon_x(z) \geqslant 0$ and $\epsilon_y(z) \geqslant 0$ are scalar and $\oslash$ designates the element wise division of two vectors. The feasible sets $d\mathcal{X}$ for (d_x, d_{s_x}) and the set-valued map that defines a feasible set $d\mathcal{Y}(d_x)$ for (d_y, d_{s_y}) are obtained from the first order linearization of the functions in $\mathcal{X}$ and $\mathcal{Y}(d_y)$ and are given by

$$\begin{aligned} d\mathcal{X} &= \{(d_x, d_{s_x}) \in \mathbb{R}^{n_x} \times \mathbb{R}^{m_x} : G_x(x) + \boldsymbol{\nabla}_x G_x(x)' d_x = \mathbf{0}, \\ &\qquad F_x(x) + s_x + \boldsymbol{\nabla}_x F_x(x)' d_x + d_{s_x} = \mathbf{0}\} \\ d\mathcal{Y}(d_x) &= \{(d_y, d_{s_y}) \in \mathbb{R}^{n_y} \times \mathbb{R}^{m_y} : G_y(x,y) + \boldsymbol{\nabla}_x G_y(x,y)' d_x + \boldsymbol{\nabla}_y G_y(x,y)' d_y \\ &\qquad = \mathbf{0}, F_y(x,y) + s_y + \boldsymbol{\nabla}_x F_y(x,y)' d_x + \boldsymbol{\nabla}_y F_y(x,y)' d_y + d_{s_y} = \mathbf{0}\}. \end{aligned}$$

If $\boldsymbol{\nabla}_x G_x(x)$ and $\boldsymbol{\nabla}_y G_y(x,y)$ have linearly independent columns, we propose to obtain $(d_x, d_{s_x}, d_y, d_{s_y})$ as the optimizers and $(d_{\nu_y}, d_{\lambda_y}, d_{\nu_x}, d_{\lambda_x})$ the associated Lagrange multipliers of the minmax optimization

$$\min_{\bar{d}_x, \bar{d}_{s_x} \in d\mathcal{X}} \quad \max_{\bar{d}_y, \bar{d}_{s_y} \in d\mathcal{Y}(\bar{d}_x)} K(\bar{d}_x, \bar{d}_{s_x}, \bar{d}_y, \bar{d}_{s_y}) \tag{3.39}$$

where $\epsilon_x(z)$ and $\epsilon_y(z)$ are chosen such that the solution to (3.39) is unique. We can apply to (3.39) the second order condition from Proposition 3.2.6 and obtain that $\epsilon_x(z)$ and $\epsilon_y(z)$ need to be chosen to satisfy

$$\begin{aligned} \mathrm{In}(J_{yy}f(z) - E_y(z)) &= (l_y{+}m_y, n_y + m_y, 0) \text{ and} \\ \mathrm{In}(J_{zz}f(z) + E(z)) &= (n_x + m_x + l_y{+}m_y, l_x + m_x + n_y + m_y, 0) \end{aligned} \tag{ConsLQAC}$$

where $E_y(z) := \mathrm{diag}(\epsilon_y(z)\mathbf{1}_{n_y}, \mathbf{0}_{l_y+2m_y})$ and $E(z) := \mathrm{diag}(\epsilon_x(z)\mathbf{1}_{n_x}, \mathbf{0}_{m_x}, -\epsilon_y(z)\mathbf{1}_{n_y}, \mathbf{0}_{l_y+2m_y+l_x+m_x})$; $J_{zz}f(z)$ is the equivalent of the matrix defined in (3.36b) for the problem (3.39) and can be shown to be equal to

$$J_{zz}f(z) = S^{-1/2}H_{zz}f(z)S^{-1/2} = S^{-1}\boldsymbol{\nabla}_z g(z,b)'. \tag{3.40}$$

with $S = \mathrm{diag}(\mathbf{1}_{n_x}, s_x, \mathbf{1}_{n_y}, s_y, \mathbf{1}_{l_y+m_y+l_x+m_x})$; $J_{yy}f(z)$ is the equivalent partition of $J_{zz}f(z)$ as $H_{yy}(z)$ is of $H_{zz}(z)$. We will call these conditions the Constrained Local Quadratic Approximation Conditions (ConsLQAC). In this case, it is straightforward to show that modifying the basic primal-dual interior-point iterations in (3.38) by taking d_z from the solution of (3.39) leads to the iterations

$$z^+ = z + \alpha d_z = z - \alpha(J_{zz}f(z) + E(z))^{-1}S^{-1}g(z,b). \tag{3.41}$$

Analogously to what was the case in unconstrained minmax optimization, choosing $\epsilon_x(z)$ and $\epsilon_y(z)$ such that the (ConsLQAC) hold is not sufficient to guarantee that P2 and P3 hold for the modified primal-dual interior-point method (a counter example can be found in Section 3.3.2). Our next theorem is the extensions of Theorem 3.2.4 to the modified primal-dual interior-point and has the equivalent consequences: For property P3 to hold, as long as $\epsilon_x(z)$ is large enough, taking $\epsilon_y(z) > 0$ will not impair the algorithm's capacity to converge towards a local minmax; this can be useful, for instance, if $\mathrm{In}(J_{zz})$ has an eigenvalue close to 0. For property P2 to hold, in order to guarantee that the modified primal-dual interior-point method cannot converge towards an equilibrium point that is not local minmax, the (ConsLQAC) are sufficient only whenever $\text{In}(J_{zz}f(z)) \neq (n_x+m_x+l_y+m_y, l_x+m_x+n_y+m_y, 0)$. Otherwise, $\epsilon_x(z)$ needs to be taken large enough such that $\text{In}(J_{zz}f(z) + \mu E(z)) \neq (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$ for some $\mu \in (0, 1)$.

Theorem 3.2.7 (Stability of modified interior-point method for minmax) *Let $\alpha = 1$ and (z, b) with $b > 0$, be an equilibrium point in the sense that $g(z, b) = \mathbf{0}$. Assume the LICQ hold at z, that $J_{zz}f(z)$ and $J_{yy}f(z)$ are invertible, and that $J_{zz}f(\cdot)$ is differentiable in a neighborhood around z. Then there exists functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ that are constant in a neighborhood around z, satisfy the* (ConsLQAC) *at z and guarantee that if:*

i) z is a local minmax, then it is a LAS equilibrium point of (3.41)*.*

ii) z is not a local minmax, then it is an unstable equilibrium point of (3.41)*.*

Proof. Let us define the partitions, $J_{xx}f(z)$, $J_{yx}f(z)$, and $J_{x\lambda}f(z)$ of $J_{zz}f(z)$ analogously to the partitions $H_{xx}f(z)$, $H_{yx}f(z)$, and $H_{x\lambda}f(z)$ of $H_{zz}f(z)$.

Using the same arguments as in the proof of Theorem 3.1.3, we conclude that the Jacobian of the dynamic system (3.41) around a point z such that $g(z, b) = \mathbf{0}$ is

$$I - \alpha\Big(J_{zz}f(z) + E(z)\Big)^{-1} S^{-1}\boldsymbol{\nabla}_z g(z, b)' = I - \alpha\Big(J_{zz}f(z) + E(z)\Big)^{-1} J_{zz}f(z) \tag{3.42}$$

Moreover from (3.40) we have that $\text{In}(H_{zz}f(z)) = \text{In}(S^{1/2}J_{zz}f(z)S^{1/2})$. Using Sylvester's law of inertia [116, Theorem 1.5], this simplifies to $\text{In}(H_{zz}f(z)) = \text{In}(J_{zz}f(z))$. If a point z is such that $g(z, b) = \mathbf{0}$, then one can check (3.37) using $J_{zz}f(z)$ and $J_{yy}f(z)$.

Let us define the matrices

$$R_y(\mu) = Z_y(z)' \begin{bmatrix} \boldsymbol{\nabla}_{yy}L(z) - \epsilon(z)\mu I & \mathbf{0} \\ \mathbf{0} & -\operatorname{diag}(\lambda_y \oslash s_y) \end{bmatrix} Z_y(z) \tag{3.43a}$$

$$R_x(\mu) = Z_x(z)'\Big(J_{xx}f(z) - J_{xy}f(z)(J_{yy}f(z) - \mu E_y(z))^{-1}J_{yx}f(x,y) + \mu E_x(z)\Big)Z_x(z) \tag{3.43b}$$

where $Z_y(z) \in \mathbb{R}^{n_y+m_y, n_y-l_y}$ and $Z_x(z) \in \mathbb{R}^{n_x+m_x, n_x-l_x}$ are any full column rank matrices such that

$$\begin{bmatrix} \boldsymbol{\nabla}_y G_y(x,y) & -\boldsymbol{\nabla}_y F_y(x,y) \\ -I & 0 \end{bmatrix} Z_y(z) = \mathbf{0} \quad \text{and} \quad J_{x\lambda}f(z)'\,Z_x(z) = \mathbf{0}. \tag{3.44}$$

Using the same reasoning as in the proof of Proposition 3.2.6 one can conclude that

$$\mathrm{In}(J_{yy}f(z) - \mu E_y(z)) = \mathrm{In}(R_y(\mu)) + (l_y + m_y, l_y + m_y, 0)$$
$$\mathrm{In}(J_{zz}f(z) + \mu E(z)) = \mathrm{In}(R_x(\mu)) + \mathrm{In}(J_{yy}f(z) - \mu E_y(z)) + (l_x + m_x, l_x + m_x, 0),$$

which implies that the (ConsLQAC) can be stated as

$$R_y(1) \prec \mathbf{0} \quad \text{and} \quad R_x(1) \succ \mathbf{0}.$$

This means that the exact same arguments used in the proof of the unconstrained minmax in Theorem 3.2.4 can be used for the constrained case. More specifically, each arguments with

$$\boldsymbol{\nabla}_{yy}f(x,y) - \epsilon_y(x,y)\mu I$$

and

$$\boldsymbol{\nabla}_{xx}f(x,y) - \boldsymbol{\nabla}_{xy}f(x,y)(\boldsymbol{\nabla}_{yy}f(x,y) - \mu\epsilon_y(x,y)I)^{-1}\boldsymbol{\nabla}_{yx}f(x,y) + \mu\epsilon_x(x,y)I.$$

has an analogous statement with $R_y(\mu)$ and $R_x(\mu)$, respectively. For the sake of completeness, we highlight the main points of the analogy.

First, when z is such that (3.37) holds, the sufficient condition for z to be a LAS equilibrium point of (3.41) is that

$$\boldsymbol{\nabla}_\mu R_x(0) = Z_x(z)'\Big(E_x(z) - J_{xy}f(z)J_{yy}f(z)^{-1}E_y(z)J_{yy}f(z)^{-1}J_{yx}f(z)\Big)Z_x(z) \geq 0. \tag{3.45}$$

The only extra argument needed is to show that condition (3.45) is always feasible for some $\epsilon_x(z)$ large enough. This is not evident as the matrix

$$M := -J_{xy}f(z)J_{yy}f(z)^{-1}E_y(z)J_{yy}f(z)^{-1}J_{yx}f(z)$$

has size $(n_x + m_x) \times (n_x + m_x)$ while $E_x(z)$ only has n_x nonzero elements in the diagonal. However, because of the structural zeros in $J_{xy}f(z)$ and $E_y(z)$, one can verify with some algebraic manipulation that $\operatorname{rank}(M) := r \leqslant \min(n_x, n_y)$. Let Λ be the matrix with eigenvalues of M in decreasing order and V its associated eigenvectors such that $M = V\Lambda V'$. We can partition V into V_1 of size (r, r) associated to the nonzero eigenvalues of M and $V_2 = I_{n_x+m_x-r}$. This partition means that $E_x(z) = V'E_x(z)V$, which means on can conclude that

$$\boldsymbol{\nabla}_\mu R_x(\mu) = Z_x(z)'V'\Big(E_x(z) + \Lambda\Big)VZ_x(z),$$

which implies that one can always take ϵ_x large enough such that for each negative diagonal entries of Λ, the equivalent diagonal element of $(E_x(z) + \Lambda)$ is positive.

Now the second part, let us prove the statement when z is such that the second order conditions in (3.37) do not hold. We need to prove that

$$J_{zz}f(z) + \mu E(z) \tag{3.46}$$

is singular for some $\mu \in (0, 1)$. On the one hand, using the same analysis as in the proof of Theorem 3.2.4, we conclude that the (ConsLQAC) are sufficient to guarantee that z is an unstable equilibrium point of (3.41) if $\operatorname{In}(J_{zz}f(z)) \neq (n_x+m_x+l_y+m_y, l_x+m_x+n_y+m_y, 0)$. On the other hand, if $\operatorname{In}(J_{zz}f(z)) = (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$, than we can guarantee that by taking ϵ_x sufficiently large, there is a $\mu \in (0, 1)$ such that $\operatorname{In}(J_{zz}f(z) + \mu E) \neq (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$, which means that z is an unstable equilibrium point of (3.41). This concludes the proof.

3.3 Algorithmic development and numerical examples

The following algorithm combines the result of the previous section to propose a method for selecting $\epsilon_x(z)$ and $\epsilon_y(z)$ that satisfies the (ConsLQAC) and guarantees the stability properties of Theorem 3.2.7. We only state the algorithm for the constrained case, its specialization to the unconstrained case is straightforward. In order to keep the algorithm more simple and to highlight the instability property, we chose to use the functions $\epsilon_y(\cdot) = \epsilon_x(\cdot) = 0$ whenever the algorithm is near a local minmax.

Algorithm 1 Primal-dual interior-point method for minmax

```
Require: An initial point $z = (x, s_x, y, s_y, \nu_y, \lambda_y, \nu_x, \lambda_x)$, an initial barrier parameter
    value $b$, a barrier reduction factor $\sigma \in (0, 1)$, a stopping accuracy $\delta_s \geqslant 0$, a $\delta_\epsilon > 0$
    that defines a neighborhood for stopping to adjust $\epsilon_x$ and $\epsilon_y$.
while $\|g(z, b)\|_\infty > \delta_s$ do
    if $\|g(z, b)\|_\infty > \delta_\epsilon$ then
        $\epsilon_x \leftarrow 0, \epsilon_y \leftarrow 0$
        if (ConsLQAC) cannot be satisfied with $\epsilon_y = \epsilon_x = 0$ then
            Increase $\epsilon_y$ until
                $$\text{In}(J_{yy}f(z) - E_y) = (l_y + m_y, n_y + m_y, 0)$$
            Increase $\epsilon_x$ until
                $$\text{In}(J_{zz}f(z) + E) = (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$$
            if $\text{In}(J_{zz}f) = (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$ then
                Increase $\epsilon_x$ until, for some value of $\mu \in (0, 1)$,
                    $$\text{In}(J_{zz}f(z) + \mu E(z)) \neq (n_x + m_x + l_y + m_y, l_x + m_x + n_y + m_y, 0)$$
            end if
        end if
    end if
    Compute a new $z$ using the equation
        $$z \leftarrow z - \alpha\Big(J_{zz}f(z) + E\Big)^{-1} S^{-1} g(z, b)$$
      where $\alpha \in (0, 1]$ is selected such that the feasibility conditions
      $\lambda_y, \lambda_x, s_y, s_x > \mathbf{0}$ hold.
    if $\|g(z, b)\|_\infty \leqslant b$ then
        $b \leftarrow \sigma\, b$
    end if
```

Proposition 3.3.1 (Construction of the modified interior-point method) *Algorithm 1 generates functions $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ that satisfy the conditions of Theorem 3.2.7 in the neighborhood of any equilibrium point z^* that satisfy the assumptions of Theorem 3.2.7.*

Proof. For each z, Algorithm 1 produces values of ϵ_x and ϵ_y that only depend on z, therefore $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ are functions. Moreover, $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ are such that either the stability condition (3.45) or the instability condition (3.46) are satisfied for each z, therefore they are satisfied in the neighborhood of any equilibrium point z^*. Finally, $\epsilon_x(\cdot)$ and $\epsilon_y(\cdot)$ are constant in a neighborhood around each equilibrium point as the values of ϵ_x and ϵ_y are not adjusted when $\|g(z,b)\|_\infty \leqslant \delta_\epsilon$.

In Algorithm 1, for each z, (ϵ_x, ϵ_y) is chosen to satisfy the conditions of Theorem 3.2.7, and therefore generate the desired stability and instability. This means that the algorithm essentially guarantees that the modified primal-dual interior-point method can only converge to an equilibrium point if such point is a local minmax. A key point of the algorithm is that it only uses the inertia of matrices, which can be efficiently computed using either the LBLt or LDLt decomposition, as we further detail in the following remark.

Remark 3.3.2 (Computing the inertia) *It is not necessary to actually compute the eigenvalues of $J_{zz}f(z)$ in order to determine the inertia. A first option is to use the lower-triangular-block-lower-triangular-transpose (LBLt) decomposition [54, Appendix A], which decomposes $J_{zz}f(z)$ into the product LBL' where L is a lower triangular matrix and B a block diagonal one, the inertia of B is the same as the inertia of $J_{zz}f(z)$.*

Let $\Gamma = \operatorname{diag}(\gamma \mathbf{1}_{n_x+m_x}, -\gamma \mathbf{1}_{n_y+m_y}, \gamma \mathbf{1}_{l_y+m_y}, -\gamma \mathbf{1}_{l_x+m_x})$, with γ a small positive number. A second approach is to use the lower-triangular-diagonal-lower-triangular-transpose (LDLt) decomposition, to decompose $J_{zz}f(z)+\Gamma$ into the product LDL' where L is a lower triangular matrix and D is a diagonal matrix; the inertia of D, which is given by the *number of positive, negative and zero elements of the diagonal of D, gives the inertia of $J_{zz}f(z) + \Gamma$. The matrix Γ introduces a distortion in the inertia but it helps to stabilize the computation of the LDLt decomposition, which tends to be faster than the LBLt decomposition. This is the approach we use in our implementation; it has been studied in primal-dual interior-point algorithms for minimization and the distortion introduced by Γ tends to be compensated by a better numerical algorithm [117, 118].* □

3.3.1 Benchmark example for unconstrained minmax

Consider the following functions

$$\begin{aligned}
f_1(x,y) &= 2x^2 - y^2 + 4xy + 4/3y^3 - 1/4y^4 \\
f_2(x,y) &= (4x^2 - (y - 3x + 0.05x^3)^2 - 0.1y^4)\exp\bigl(-0.01(x^2+y^2)\bigr) \\
f_3(x,y) &= (x - 0.5)(y - 0.5) + \exp\bigl(-(x-0.25)^2 - (y-0.75)^2\bigr) \\
f_4(x,y) &= xy.
\end{aligned}$$

The first three have been used as examples in [69,73,119] respectively, whereas the fourth one is a well known case for a simple but challenging function to find the local minmax. These problems all satisfy the assumption of Theorem 3.2.4 and have local minmax points. We have chosen these functions because, as we will show, they illustrate some interesting behaviors.

Our goal is to compare the performance of Algorithm 1 to the performance of two well established algorithms. On the one hand, we look at the performance of a "pure" Newton algorithm, *i.e.* using $\epsilon_x(\cdot) = \epsilon_y(\cdot) = 0$. On the other hand, we look into the convergence of a Gradient Descent Ascent (GDA), *i.e.*

$$\begin{aligned}
x^+ &= x - \alpha_x \nabla_x f(x,y) \\
y^+ &= y + \alpha_y \nabla_y f(x,y)
\end{aligned}$$

	Pure Newton			GDA			Algorithm 1		
	Cnvg	Cnvg mm	Iter	Cnvg	Cnvg mm	Iter	Cnvg	Cnvg mm	Iter
f_1	1000	1000	4.1	1000	1000	485	1000	1000	5.7
f_2	1000	665	7.3	976	976	18195	996	996	8.1
f_3	954	485	4.8	373	373	40936	709	709	7.1
f_4	1000	1000	1	0	0	–	1000	1000	1

Table 3.1: Comparing the performance of Pure Newton's method, Gradient Descent Ascent and Algorithm 1

.

where α_x and α_y are constant and different for each problem; we did our best to select the best values α_x and α_y for each problem.

Each algorithm is initialized 1000 times, using the same initialization for the three of them each time. We compare their convergence properties according to three criteria: the number of times the algorithm converged (Cnvg), the number of times it converged to a local minmax point (Cnvg mm) and the average number of iterations to converge to a local minmax point (Iter). The algorithm is terminated when the infinity norm of the gradient is smaller than $\delta_s = 10^{-5}$ and we declare that they did not converge if it has not terminated in less than 500 iterations for the pure Newton and Algorithm 1, and 50 000 for GDA. The result of the comparison is displayed in Table 3.1. The key take away from these examples is that Algorithm 1 never converges towards an equilibrium point that is not a local minmax, in contrast with the pure Newton method which is attracted to any equilibrium point. Here is a detailed observation from this comparison.

- The pure Newton algorithm has good overall convergence for all the problems, but it also tends to often converge towards an equilibrium point that is not a local minmax problems. On the other hand, the pure Newton converges to a local minmax in less iterations than the other two methods. While this is expected when comparing to the GDA, it might not be clear why it is the case when compared to Algorithm 1. We believe the most likely reason is that by taking ϵ_x and ϵ_y different than 0, it requires some more iterations to converge towards a local minmax.

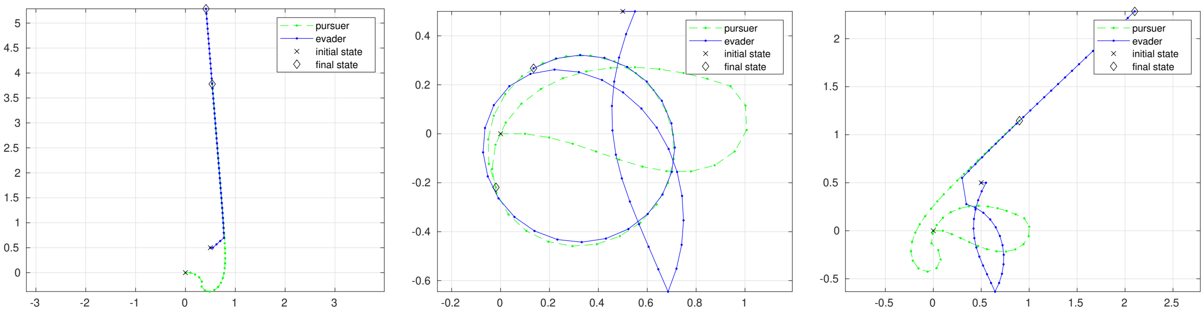
- The GDA algorithm seems to enjoy the property of always converging towards a local minmax, and except for $f_3(\cdot)$ and $f_4(\cdot)$, it has good rate of convergence. However, GDA takes an exceptionally long number of iterations to converge. This is somehow expected from the fact that it is a first order method, and it is partially compensated by each iteration being more simple to compute. However, one must keep in mind that none of this takes into account the time that needs to be spent adjusting the step sizes until a good convergence rate can be obtained.

- At last, Algorithm 1 is across the board the algorithm with better convergence towards local minmax, and it does so in the smallest number of iterations. As it was expected from the theory, Algorithm 1 never converges towards an equilibrium point that is not a local minmax. From a numerical perspective, the biggest takeaway is that while our results are only about local convergence, the algorithm still enjoys good global convergence properties; only in $f_3(\cdot)$ it does not converge essentially 100% of the time.

- Function $f_4(\cdot)$ is particularly interesting example. First, notice that the pure Newton converges in one iteration. This is expected as the iterations are given by

$$\begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

  This is in stark contrast with GDA which, as it is well known, does not converge towards the local minmax. As for Algorithm 1, it converges even though it does



(a) While guaranteeing instability

(b) Without guaranteeing instability

(c) With guaranteeing instability only after $t = 25$

Figure 3.1: Trajectory for Homicidal Chauffeur problem with and without guaranteeing instability at equilibrium points that are not a local minmax.

not satisfy the assumptions of Theorem 3.2.4, further emphasizing that these are sufficient but not necessary conditions. Notice that Algorithm 1 is not the same as the pure Newton as the Hessian will be modified with an $\epsilon_y(x, y) > 0$ to guarantee that the portion of the Hessian associated to the maximization is negative definite.

3.3.2 The homicidal chauffeur example for constrained minmax

In the homicidal chauffeur problem, a pursuer driving a car is trying to hit a pedestrian, who (understandably) is trying to evade it. The pursuer is modeled as a discrete time Dubins's vehicle with equations

$$x_p^+ = \begin{bmatrix} x_p^{(1)} + v \cos x_p^{(3)} \\ x_p^{(2)} + v \sin x_p^{(3)} \\ x_p^{(3)} + u \end{bmatrix} =: \phi_p(x_p, u)$$

where $x_p^{(i)}$ designates the i$^{\text{th}}$ element of the vector x_p, v is a constant forward speed and u is the steering, over which the driver has control. The pedestrian is modeled by the accumulator

$$x_e^+ = x_e + d =: \phi_e(x_e, d)$$

where d is the velocity vector. Given a time horizon T, and initial positions $x_e(t)$ and $x_p(t)$, we want to solve

$$\min_{U \in \mathcal{U}} \quad \max_{D \in \mathcal{D}} \sum_{i=0}^{T-1} \left\| x_p^{(1,2)}(t+i+1) - x_e(t+i+1) \right\|_2^2 + \gamma_u u(t+i)^2 - \gamma_d \|d(t+i)\|_2^2 \tag{3.47}$$

where $x_p^{(1,2)}$ designates the first and second elements of the vector x_p; γ_u and γ_d are positive weights; and U, $\mathcal{U}$, D and $\mathcal{D}$ are defined for $i = 0, \ldots, T-1$

$$U := u(t+i), x_p(t+i+1)$$

$$\mathcal{U} := \{u(t+i), x_p(t+i+1) : \ u(t+i)^2 \leqslant u_{max}^2, x_p(t+i+1) = \phi_p\big(x_p(t+i), u(t+i)\big)\}$$

$$D := d(t+i), x_e(t+i+1)$$

$$\mathcal{D} := \{d(t+i), x_e(t+i+1) : \ \|d(t+i)\|_2^2 \leqslant d_{max}^2, x_e(t+i+1) = \phi_e\big(x_e(t+i), d(t+i)\big)\}.$$

Instead of explicitly computing the solution of the trajectory of the pursuer and evaders, we are implicitly computing them by setting the dynamics as equality constraints; we will show shortly that this has an important impact on the scalability of the algorithm.

Each player is controlled using Model Predictive Control (MPC), meaning that at each time step t we solve (3.47) obtaining controls $u(t)$ and $d(t)$, which are then used to control the system for the next time step. The problem satisfy the assumptions of Theorem 3.2.7, as it is differentiable and has local minmax points for which the LICQ and strict complementarity hold.

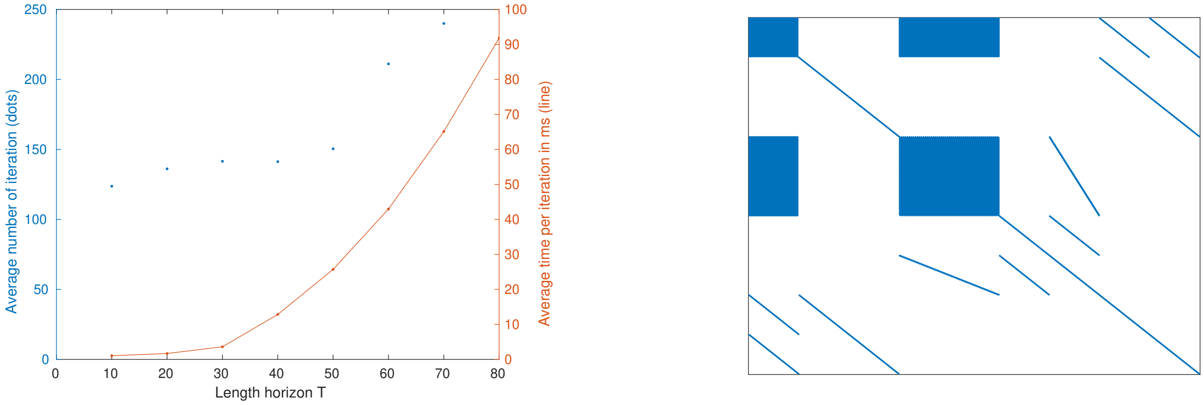
The importance of guaranteeing instability It is natural to ask whether it is important to enforce the instability guarantee, specially in the case where the (ConsLQAC) is not enough, meaning one needs to use line 7 of Algorithm 1. In Figure 3.1 we show what can happen if they are not enforced. We take the homicidal chauffeur problem with a horizon of $T = 20$ and we run the MPC control for $t = 1, \ldots, 50$. In one case we enforce the instability guarantee, meaning that we use line 7 of Algorithm 1, on the second case we only enforce the (ConsLQAC), and on the third case we only enforce the instability guarantees after $t = 25$. In all cases, we start the system with the exact same initial conditions.

In the first case, the evader (which is the maximizer), is able to find a control that allows it to get further from the pursuer. The average cost for all the time steps ($t = 1, \ldots, 50$) ends up being around 0.2. In the second case, the solver keeps being attracted towards a point that is not a local minmax (and more precisely, not a local maximum), which means that the evader is not capable of escaping the pursuer; as a consequence, the average cost for all the time steps ends up being around 0.05, which is lower, as expected. Finally, in the third case, at $t = 25$ the solver starts to be able to converge towards a local minmax, and the evader is able to escape from the pursuer.

This example illustrates how crucial it is to enforce instability. By doing it, we guarantee that the algorithm can only converge towards an equilibrium point that is a local minmax, and this can completely change the numerical solution.

Exploiting sparsity Instead of setting the dynamics as equality constraints in (3.47), one could simply find the solution of the trajectory equation at each time step. This means to explicitly calculate $x_p(t+i+1) = \phi_p\Big(\phi_p\big(\ldots, u(t+i-1)\big), u(t+i)\Big)$. In the MPC literature, this is known as the sequential approach, versus the simultaneous approach we used in (3.47) [120, Chapter 8.1.3]. We want to study the scalability of the algorithm by enlarging the horizon T, both when using the sequential and the simultaneous approaches.

The sequential approach solves an optimization problem in a smaller space state, because it only needs to solve the optimization for $u(t), \ldots, u(t+T)$ and $d(t), \ldots, d(t+T)$ and it does not have to handle equality constraints. However, as we can see from



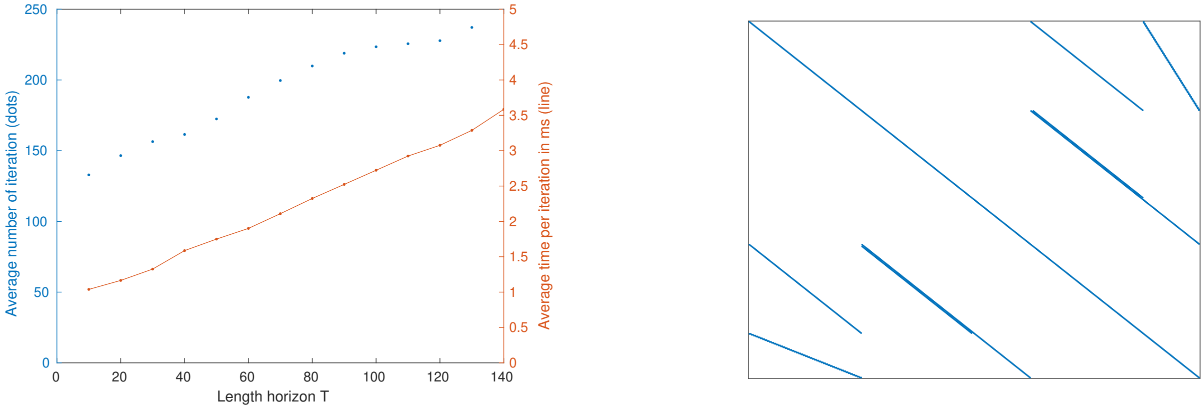
(a) Computational scaling for solving homicidal chauffeur per horizon length

(b) Structural sparsity pattern of $J_{zz}f(z)$

Figure 3.2: Scaling of homicidal chauffeur with horizon length and sparsity pattern of the Hessian when using the sequential approach

the sparsity pattern in Figure 3.2b, the Hessian is rather dense, with large parts of it containing nonzero entries. As it can be seen in Figure 3.2a, the algorithm scales rather poorly as the horizon length (and hence, the number of variables) increases; it no longer converges reliably after $T = 80$.

The simultaneous approach on the other hand solves the optimization problem in a much larger space state, because not only it needs to also solve for $u(t), \dots, u(t+T)$ and $d(t), \dots, d(t+T)$, but also for $x_p(t), \dots, x_p(t+T)$ and $x_e(t), \dots, x_e(t+T)$ and it also needs to handle equality constraints. Fortunately, as we can see from the sparsity pattern in Figure 3.3b, most of the entries in the Hessian are actually structurally zero (meaning they are always zero). `TensCalc`'s implementation of the LDLt factorization exploits sparsity patterns and scales roughly in $O(T)$, which makes it substantially more efficient than standard LDLt decomposition, which scales in $O(T^3)$ [54, Appendix A]. At each step of Algorithm 1, most of the time is spent computing the LDLt decomposition, either for adjusting ϵ_x and ϵ_y or to invert $H_{zz}f(z)$. As a consequence, we can see in Figure 3.3a that both the number of iterations necessary to solve the optimization as



(a) Computational scaling for solving homicidal chauffeur per horizon length

(b) Structural sparsity pattern of $J_{zz}f(z)$

Figure 3.3: Scaling of homicidal chauffeur with horizon length and sparsity pattern of the Hessian

well as the time per iteration scale roughly linear, the first being multiplied by about 1.7 while the second by 3.5 while the horizon length T is multiplied by roughly 30.

Remark 3.3.3 (Minmax problems with shared dynamics) *In the homicidal chauffeur, the control of the pursuer does not impact the* dynamics *of the evader, and vice versa. This is why in* (3.47) *the dynamics can be set as equality constraints independently for the min and for the max.*

Now consider the problem

$$x^+ = f(x, u, d)$$

where u is the control and d is the disturbance and one wants to minimize a cost function $V(x(1), \dots, x(T), u(0), \dots, u(T-1))$ given the worst disturbance $d(1), \dots, d(T)$. Because both the control and the disturbances influence the dynamics, we need to include the dynamics as equality constraints for the maximization, *leading to the optimization problem*

$$\min_{u(i)\in\mathcal{U}, i=0,\dots,T-1} \max_{\substack{d(i)\in\mathcal{D}, x(i+1), i=0,\dots,T-1: \\ x(i+1)=f(x(i),u(i),d(i))}} V\Big(x(1), \dots, x(T), u(0), \dots, u(T-1)\Big)$$

where $\mathcal{U}, \mathcal{D}$ are the feasible sets for the control and disturbances. It is important to notice that x just acts as a latent/dummy variable that allows us to avoid solving the trajectory equation. Setting it as a maximization variable does not changes the result as x is always exactly determined by the value of u and d. It does, however, improves the numerical efficiency of the algorithm as now the Hessian matrices are sparse and their LDL decomposition can be efficiently computed. □

3.4 Conclusion

The main contribution of this article is the construction of Newton and primal-dual interior-point algorithm for nonconvex-nonconcave minmax optimization that can only converge towards an equilibrium point if such point is a local minmax. We established this results by modifying the Hessian matrices such that the update steps can be seen as the solution of quadratic programs that locally approximate the minmax problem. While our results are only local, using numerical simulations we see that the algorithm is able to make progress towards a solution even if it does not start close to it. We also illustrated using numerical examples how important it is to have a formulation of the minmax problem such that the Hessian matrix is sparse.

The main future direction would be to develop non-local convergence results. We believe that the best approach to obtain such results would be to develop a type of Armijo rule which could be used to obtain similar results to those from minimization. Developing filters and merit function could also play an important role in coming up with ways to improve the algorithm's convergence.

3.A Appendix of Chapter 3

3.A.1 Proof of Proposition 3.1.5 (constrained minimization)

The first step is to show that $g(z,0) = \mathbf{0}$ is equivalent to the Karush–Kuhn–Tucker (KKT) conditions [54, Chapter 12]. Consider the “full" Lagrangian $\tilde{L}(x, s_x, \nu_x, \lambda_x, \tau_x) = f(x) + \nu_x' G_x(x) + \lambda_x'(F_x(x) + s_x) - \tau_x' s_x$ for the optimization (3.9). The KKT condition would then be that

$$\begin{bmatrix} \boldsymbol{\nabla}_x \tilde{L}(x, s_x, \nu_x, \lambda_x, \tau_x) \\ \boldsymbol{\nabla}_{s_x} \tilde{L}(x, s_x, \nu_x, \lambda_x, \tau_x) = \lambda_x - \tau_x \\ G_x(x) \\ F_x(x) + s_x \\ \tau_x \odot s_x \end{bmatrix} = \mathbf{0} \tag{3.48}$$

and $s_x, \tau_x \geqslant \mathbf{0}$. The second equation can be used to substitute τ_x by λ_x, which gives the equality $g(z,0) = \mathbf{0}$.

Now the second order sufficient conditions. Let us start by rewriting the minimization (3.1) but instead of using as slack variables s_x with the constraint $s_x \geqslant 0$, using the slack variable $w_x \odot w_x$ (where $\odot$ is the element wise product):

$$\min_{x, w_x : G_x(x) = \mathbf{0}, F_x(x) + w_x \odot w_x = \mathbf{0}} f(x). \tag{3.49}$$

Consider now the solution cone

$$\mathcal{C}_x(z) := \{(d_x, d_w) \in \mathbb{R}^{n_x + m_x} \backslash \{\mathbf{0}\} : \boldsymbol{\nabla}_x G_x(x)' d_x = \mathbf{0}, \boldsymbol{\nabla}_x F_x(x) d_x + 2\,\mathrm{diag}(w_x) d_w = \mathbf{0}\}$$

Let $(x, w_x, \nu_x, \lambda_x)$ be a point such that the KKT conditions for (3.49) hold. As, by assumption, the LICQ and strict complementarity conditions hold, if

$$\begin{bmatrix} d_x \\ d_w \end{bmatrix}' \begin{bmatrix} \boldsymbol{\nabla}_{xx} L(z) & \mathbf{0} \\ \mathbf{0} & 2\,\mathrm{diag}(\lambda_x) \end{bmatrix} \begin{bmatrix} d_x \\ d_w \end{bmatrix}' > 0 \ \forall (d_x, d_w) \in \mathcal{C}_x(z) \tag{3.50}$$

then $(x, w_x, \nu_x, \lambda_x)$ is a local minimum of (3.49). The proof can be found in [54, Theorem 12.5].

We now need to prove that (3.50) is equivalent to the condition (3.12) from the proposition. Because the LICQ and strict complementarity hold, the set $\mathcal{C}_x(z)$ is given by the null space (a.k.a. the kernel) of the matrix

$$\tilde{H}_{x\lambda} f(z) = \begin{bmatrix} \nabla_x G_x(x) & \nabla_x F_x(x) \\ 0 & 2\operatorname{diag}(w_x). \end{bmatrix} \tag{3.51}$$

This result can be found in [54, Chapter 12.5], in the subsection "Second-order conditions and projected Hessian". Let $Z_x \in \mathbb{R}^{n_x+m_x, n_x+m_x-m_x-l_x}$ be a matrix with full column rank such that $\tilde{H}_{x\lambda} f(z)' \, Z_x = \mathbf{0}$. Then, the condition (3.50) can be rewritten as

$$Z_x' \begin{bmatrix} \nabla_{xx} L(z) & \mathbf{0} \\ \mathbf{0} & 2\operatorname{diag}(\lambda_x) \end{bmatrix} Z_x \succ 0$$

which is equivalent to say that

$$\operatorname{In}\left(Z_x' \begin{bmatrix} \nabla_{xx} L(z) & \mathbf{0} \\ \mathbf{0} & 2\operatorname{diag}(\lambda_x) \end{bmatrix} Z_x \right) = (n_x - l_x, 0, 0)$$

Now consider the matrix

$$\tilde{H}_{zz} f(z) = \begin{bmatrix} \nabla_{xx} L(z) & \mathbf{0} & \nabla_x G_x(x) & \nabla_x F_x(x) \\ \mathbf{0} & 2\operatorname{diag}(\lambda_x) & \mathbf{0} & 2\operatorname{diag}(w) \\ \nabla_x G_x(x)' & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \nabla_x F_x(x)' & 2\operatorname{diag}(w) & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{3.52}$$

As the LICQ conditions hold, according to [54, Theorem 16.3]

$$\operatorname{In}(\tilde{H}_{zz} f(z)) = \operatorname{In}\left(Z_x' \begin{bmatrix} \nabla_{xx} L(z) & \mathbf{0} \\ \mathbf{0} & 2\operatorname{diag}(\lambda_x) \end{bmatrix} Z_x \right) + (l_x + m_x, l_x + m_x, 0).$$

Therefore (3.50) holds if and only if $\text{In}(\tilde{H}_{zz}f) = (n_x + m_x, l_x + m_x, 0)$.

We have almost finished the proof, we now just need to prove that $\text{In}(\tilde{H}_{zz}f(z)) = \text{In}(H_{zz}f(z))$. Using the equality condition $F_x(x) + w_x \odot w_x = 0$, we obtain the relation $w_x = (-F_x(x))^{1/2} = s_x^{1/2}$. If we substitute back this result in $\tilde{H}_{zz}f(z)$ we almost have that $\tilde{H}_{zz}f(z)$ is equal to $H_{zz}f(z)$ except for the 2 in front of $\text{diag}(\lambda_x)$ and $\text{diag}(s^{1/2})$. Take the matrix Ξ defined by

$$\Xi = \text{diag}([\mathbf{1}_{n_x}, [a^{(1)}, a^{(2)}, ..., a^{(m_x)}], \mathbf{1}_{l_x+m_x}])$$

where

$$a^{(i)} = \begin{cases} \frac{1}{2} & \text{if } \lambda_x^{(i)} = 0 \text{ and } s_x^{(i)} \neq 0 \\ \frac{1}{\sqrt{2}} & \text{if } \lambda_x^{(i)} \neq 0 \text{ and } s_x^{(i)} = 0 \end{cases}$$

with $\lambda_x^{(i)}$ and $s_x^{(i)}$ denoting the i^{th} elements of λ_x and s_x. Then $\Xi\tilde{H}_{zz}f(z)\Xi = H_{zz}f(z)$ which, according to Sylvester's law of inertia [116, Theorem 1.5], implies that $\text{inertia}(\tilde{H}_{zz}f(z)) = \text{inertia}(H_{zz}f(z))$, which finishes the proof. □

3.A.2 Proof of Proposition 3.2.6 (constrained minmax optimization)

First, using the exact same reasoning as in the proof of Proposition 3.1.5, one can show that $g(z, 0) = 0$ is equivalent to the first order necessary condition in [53].

Similarly to what we did in the proof of Proposition 3.1.5, let us start by rewriting the constrained minmax optimization (3.35) using the slack variables $w \odot w$:

$$\min_{x,w_x:G_x(x)=\mathbf{0},F_x(x)+w_x\odot w_x=\mathbf{0}} \quad \max_{y,w_y:G_y(x,y)=\mathbf{0},F_y(x,y)+w_y\odot w_y=\mathbf{0}} f(x, y).$$

Consider the solution cones

$$\mathcal{C}_y(z) := \{(d_y, d_{w_y}) \in \mathbb{R}^{n_y+m_y}\backslash\{\mathbf{0}\} : \boldsymbol{\nabla}_y G_y(x, y)d_y = \mathbf{0}, \boldsymbol{\nabla}_y F(x, y)d_y + 2\,\text{diag}(w_y)d_{w_y} = \mathbf{0}\}$$

and

$$\mathcal{C}_x(z) := \{(d_x, d_{w_x}) \in \mathbb{R}^{n_x+m_x} \backslash \{\mathbf{0}\} : \boldsymbol{\nabla}_x G_x(x)' d_x = \mathbf{0} \boldsymbol{\nabla}_x F_x(x) d_x + 2\operatorname{diag}(w_x) d_{w_x} = \mathbf{0}\}$$

Let z be a point such that $g(z, 0) = \mathbf{0}$. As, by assumption, the LICQ and strict complementarity hold, if

$$\begin{bmatrix} d_y \\ d_{w_y} \end{bmatrix}' \begin{bmatrix} \boldsymbol{\nabla}_{yy} L(z) & \mathbf{0} \\ \mathbf{0} & -2\operatorname{diag}(\lambda_y) \end{bmatrix} \begin{bmatrix} d_y \\ d_{w_y} \end{bmatrix} < 0 \ \forall \ (d_y, d_{w_y}) \in \mathcal{C}_y(z) \tag{3.53a}$$

and

$$\begin{bmatrix} d_x \\ d_{w_x} \end{bmatrix}' \Big(H_{xx} L(z) - H_{xy} f(z) H_{yy} f(z)^{-1} H_{xy} f(z)' \Big) \begin{bmatrix} d_x \\ d_{w_x} \end{bmatrix} > 0 \ \forall \ (d_x, d_{w_x}) \in \mathcal{C}_x(z) \tag{3.53b}$$

then $(x, w_x, \nu_x, \lambda_x)$ is a local minimum of (3.49). The proof can be found in [53, Theorem 3.2].

The proof between the equivalence of the condition (3.53a) and $\operatorname{In}(H_{yy} f(z)) = (l_y + m_y, n_y + m_y, 0)$ is almost identical to the proof of Proposition (3.1.5).

The condition on the inertia of $\operatorname{In}(H_{zz} f(z))$ require some more development. In an analogous way to the proof of Proposition (3.1.5), let Z_x be a matrix with full column rank such that $H_{x\lambda} f(z)' Z_x = \mathbf{0}$. Then the sufficient conditions (3.53b) for the reformulated outer minimization is

$$Z_x' \Big(H_{xx} f(z) - H_{yx} f(z)' H_{yy} f(z)^{-1} H_{yx} f(z) \Big) Z_x \succ 0. \tag{3.54}$$

We want now to define a new partition of $H_{zz} f(z)$ which we will use to finish the proof. Consider the matrices

$$\bar{H}_{zz} f(z) = \begin{bmatrix} H_{xx} f(z) & H_{xy} f(z) \\ H_{xy} f(z)' & H_{yy} f(z) \end{bmatrix} \quad \text{and} \quad \bar{H}_{x\lambda} f(z) = \begin{bmatrix} H_{x\lambda} f(z) \\ \mathbf{0}_{n_y+m_y+l_y+m_y, l_x+m_x} \end{bmatrix}.$$

such that

$$H_{zz}f(z) = \begin{bmatrix} \bar{H}_{zz}f(z) & \bar{H}_{x\lambda}f(z) \\ \bar{H}_{x\lambda}f(z)' & \mathbf{0}_{l_x+m_x} \end{bmatrix}$$

Let the matrix

$$\bar{Z}_x := \begin{bmatrix} Z_x & \mathbf{0}_{n_x+m_x,n_y+m_y+l_y+m_y} \\ \mathbf{0}_{n_y+m_y+l_y+m_y,n_x-l_x} & I_{n_y+m_y+l_y+m_y}. \end{bmatrix}$$

One can show that $\bar{Z}_x$ is full column rank and such that $\bar{H}_{x\lambda}f(z)'\,\bar{Z}_x = \mathbf{0}$. Therefore if we apply [54, Theorem 16.3] to $H_{zz}f(z)$ (with the new partitioning) gives

$$\operatorname{In}(H_{zz}f(z)) = \operatorname{In}\left(\bar{Z}_x'\bar{H}_{zz}f(z)\bar{Z}_x\right) + (l_x+m_x, l_x+m_x, 0)$$

In turn, $\operatorname{In}\left(\bar{Z}_x'\bar{H}_{zz}f(z)\bar{Z}_x\right)$ can be simplified using Haynsworth inertia additivity formula [116, Theorem 1.6]:

$$\begin{aligned} &\operatorname{In}\left(\bar{Z}_x'\bar{H}_{zz}f(z)\bar{Z}_x\right) \\ &= \operatorname{In}\left(\begin{bmatrix} Z_x'H_{xx}f(z)Z_x & Z_x'H_{xy}f(z) \\ H_{xy}f(z)'Z_x & H_{yy}f(z) \end{bmatrix}\right) \\ &= \operatorname{In}\Big(Z_x'\Big(H_{xx}f(z) - H_{xy}f(z)H_{yy}f(z)^{-1}H_{xy}f(z)'\Big)Z_x\Big) + \operatorname{In}(H_{yy}f(z)). \end{aligned}$$

Therefore, if (3.53a) holds, (3.53b) is equivalent to

$$\operatorname{In}(H_{zz}f(z)) = (n_x-l_x, 0, 0) + (l_y+m_y, n_y+m_y, 0) + (l_x+m_x, l_x+m_x, 0)$$

which finishes the proof. □

Chapter 4

On the Asymptotic Convergence of Full LOLA

A fundamental question of minmax optimization is, given a pair of current points, how to pick update directions such that the new points are closer to the solution of the problem. In the previous chapter, these directions were obtained from a modified second order linearization of the minmax around the current point. The limitation of this approach is that it requires computing many LDL decomposition of the matrices, which can be unfeasible in large scale problems.

In this chapter, we take a different approach, and look at the question of obtaining minmax directions from using the concept of full descent ascent direction, which we introduce in Section 4.1. In its essence, a full descent ascent direction is a pair of vectors that updated the current values of the minimizer and maximizer such that they are aware of each other update. We show that this approach has some more fundamental connection with minmax optimization then other types of approaches.

Given the definition of full descent ascent, the next question we address is *how to* obtain a full descent ascent direction, in Section 4.2. In the method we propose, the maximizer computes the direction after the minimizer, while the minimizer computes their direction based on a gradient approximation of the step the maximizer will take. We then go on to show that this method to obtain directions produce full descent ascent directions.

Finally, in Section 4.3, we study what are the conditions such that the limit points of the generated sequence are equilibrium points, which is know as asymptotic convergence. We prove that for two types of choices of step sizes, either with fixed step sizes or using an Armijo Rule, we can obtain asymptotic convergence. Moreover, we also show a version of the capture theorem, which implies that there is a neighborhood around equilibrium points that attract the full descent ascent iterations.

Notation: The set of real numbers is denoted by $\mathbb{R}$. Given a vector $v \in \mathbb{R}^n$, its transpose is denoted by v'. Consider a differentiable function $f : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^p$. The Jacobian (or gradient if $p = 1$) at a point $(\bar{x}, \bar{y})$ according to the x variable is a matrix of size $n \times p$ and is denoted by $\boldsymbol{\nabla}_x f(\bar{x}, \bar{y})$. The partial derivative according to the coordinate x is a matrix of size $n \times p$ and is denoted by $\partial_x f(\bar{x}, \bar{y})$. Given a differentiable function $g : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\boldsymbol{\nabla}_x f(\bar{x}, g(\bar{x})) = \partial_x f(\bar{x}, g(\bar{x})) + \boldsymbol{\nabla}_x g(\bar{x}) \partial_y f(\bar{x}, g(\bar{x}))$. For a twice differentiable function, the cross derivative is given by $\boldsymbol{\nabla}_{xy} f(\bar{x}, \bar{y}) = \boldsymbol{\nabla}_x (\boldsymbol{\nabla}_y f(\bar{x}, \bar{y}))$.

4.1 Problem statement

Consider two non-empty, closed and convex sets [1] $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$ and a function $f : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$. The minmax optimization

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) \tag{4.1}$$

[1] We remind the reader that $\mathbb{R}^n$ is closed and convex.

denotes the problem of finding a point $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ such that $\forall y \in \mathcal{Y}$ and $\forall x \in \mathcal{X}$

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant \max_{\tilde{y} \in \mathcal{Y}} f(x, \tilde{y}).$$

If such point exists, it is called a global minmax of $f(\cdot)$.

4.1.1 Local minmax

Except in some specific cases, such as when $f(\cdot)$ is convex in x and concave in y, finding a global minmax is extremely challenging. An alternative is to look for a local minmax, which was first defined in [32].

Definition 4.1.1 (Local minmax according to Jin et al.) *A point (x^*, y^*) is said to be a local minmax of $f(\cdot)$ if there exist $\delta_0 > 0$ and a positive function $h(\cdot)$ satisfying $h(\delta) \to 0$ as $\delta \to 0$, such that for any $\delta \in (0, \delta_0]$, $\forall x \in \mathcal{X} : \|x - x^*\| \leqslant \delta$ and $\forall y \in \mathcal{Y} : \|y - y^*\| \leqslant h(\delta)$ we have that*

$$f(x^*, y) \leqslant f(x^*, y^*) \leqslant \max_{\tilde{y} \in \mathcal{Y}: \|\tilde{y} - y^*\| \leqslant h(\delta)} f(x, \tilde{y}) \qquad \square$$

Essentially, a local minmax is defined by properties that hold on neighborhoods around (x^*, y^*). Local properties have the advantage that they tend to be easier to verify than global ones. Unfortunately, a global minmax might not be a local minmax, and we refer the reader to the original paper for a counter example and an analysis on this question.

Despite this evident drawback in the definition of local minmax, one of its main advantages is that one can deduce first order necessary conditions of optimality. We state the result in a slightly more general form than it is stated in [32] in order to take into account constraints.

Proposition 4.1.2 (First order necessary condition) *Assume $f(\cdot)$ is continuously differentiable and (x^*, y^*) is a local minmax. Then, $\forall y \in \mathcal{Y}$, $(y - y^*)' \boldsymbol{\nabla}_y f(x^*, y^*) \leqslant 0$. Moreover, $\exists \delta_0 > 0$ such that $\forall x \in \mathcal{X} : \|x - x^*\| < \delta_0$, $(x - x^*)' \boldsymbol{\nabla}_x f(x^*, y^*) \geqslant 0$.* $\square$

Proof. Starting with the max, fix any $y \in \mathcal{Y}$ and let us denote $p_y := (y - y^*)$. Because $\mathcal{Y}$ is convex, for any $\beta \in [0,1]$, $y^* + \beta p_y \in \mathcal{Y}$. As (x^*, y^*) is a local maximum, there exist $\tilde{\beta} : \forall \beta \in [0, \tilde{\beta}]$ the following inequality holds

$$0 \geqslant \frac{f(x^*, y^* + \beta p_y) - f(x^*, y^*)}{\beta}$$

Because $f(\cdot)$ is continuously differentiable, according to the mean value Theorem, there exist $\bar{\beta} \in [0, \beta]$ such that the previous inequality is equivalent to

$$0 \geqslant \partial_y f(x^*, y^* + \bar{\beta} p_y)' p_y.$$

Taking the limit as β goes to 0 finishes the first part of the proof. Now for the min, take the δ_0 from the definition of local minmax, fix any $x \in \mathcal{X} : \|x - x^*\| < \delta_0$ and let us denote $p_x := (x - x^*)$. Because $\mathcal{X}$ is convex, for any $\alpha \in [0,1]$, $x^* + \alpha p_x \in \mathcal{X}$ and $\|p_x\| < \delta_0$. Take the function $h(\cdot)$ from the definition of local minmax, and define the local optimum

$$p_y^*(\alpha p_x) = \underset{p_y : y^* + p_y \in \mathcal{Y}, \|p_y\| < h(\alpha p_x)}{\arg\max}.$$

By the definition of $h(\cdot)$ we have that $p_y^*(\alpha p_x) \to 0$ as $\alpha \to 0$. Then, as (x^*, y^*) is a local minmax,

$$\begin{aligned}
0 &\leqslant \frac{f(x^* + \alpha p_x, y^* + p_y^*(\alpha p_x)) - f(x^*, y^*)}{\alpha} \\
&= \frac{f(x^* + \alpha p_x, y^* + p_y^*(\alpha p_x)) - f(x^*, y^*) + f(x^*+, y^* + p_y^*(\alpha p_x)) - f(x^*+, y^* + p_y^*(\alpha p_x))}{\alpha} \\
&\leqslant \frac{f(x^* + \alpha p_x, y^* + p_y^*(\alpha p_x)) - f(x^*, y^* + p_y^*(\alpha p_x))}{\alpha} \\
&= \partial_x f(x^* + \bar{\alpha} p_x, y^* + p_y^*(\alpha p_x))' p_x \quad \text{for some } \bar{\alpha} \in [0, \alpha]
\end{aligned}$$

taking the limit as α goes to zero finishes the proof. ■

Corollary 4.1.3 (Unconstrained conditions) *Assume $f(\cdot)$ is continuously differentiable and (x^*, y^*) is a local minmax and an interior point of $\mathcal{X} \times \mathcal{Y}$. Then $\boldsymbol{\nabla}_y f(x^*, y^*) = 0$ and $\boldsymbol{\nabla}_x f(x^*, y^*) = 0$.* □

Proof. For the max, if y^* is an interior point of $\mathcal{Y}$, then for any $y \in \mathcal{Y}$ there is a $\beta \in (0,1]$ such that $y^* - \beta(y - y^*) \in \mathcal{Y}$. Therefore we have that $(y - y^*)'\nabla_y f(x^*, y^*) \leqslant 0$ and that $-\beta(y - y^*)'\nabla_y f(x^*, y^*) \leqslant 0$ which implies that $\nabla_y f(x^*, y^*) = 0$. The proof for the min is equivalent. ■

4.1.2 Descent ascent algorithms

Consider two arbitrary functions $d_x(x,y)$ and $d_y(x,y)$ that satisfy the conditions that $x + d_x(x,y) \in \mathcal{X}$ and $y + d_y(x,y) \in \mathcal{Y}$ and a sequence of scalars $\{(\alpha^k, \beta^k)\}$ with $\alpha^k, \beta^k \in (0,1]$. Given an initial point $(x^0, y^0) \in \mathcal{X} \times \mathcal{Y}$, the sequence $\{(x^k, y^k)\}$ is recursively defined by

$$\begin{aligned} x^{k+1} &= x^k + \alpha^k d_x(x^k, y^k) \\ y^{k+1} &= y^k + \beta^k d_y(x^{k+1}, y^k). \end{aligned} \tag{4.2}$$

In general, there are no closed form expressions to obtain local minmax points. Instead, one uses descent ascent algorithms, in which one designs numerical functions $d_x(x,y)$ and $d_y(x,y)$ and sequences $\{(\alpha^k, \beta^k)\}$ such that every limit point of the sequence $\{(x^k, y^k)\}$ satisfies the first order optimality conditions of Proposition 4.1.2; **we call such points of stationary points**. The most common type of descent ascent algorithms uses alternating descent ascent sequences, for which we give the following definition:

Definition 4.1.4 (Alternating descent ascent) *We say that the sequence defined by* (4.2) *is an alternating descent ascent sequence if it satisfies*

$$f(x^{k+1}, y^k) \leqslant f(x^k, y^k) \tag{4.3a}$$

$$f(x^{k+1}, y^{k+1}) \geqslant f(x^{k+1}, y^k). \tag{4.3b}$$

with at least one of the inequalities holding strictly. By extension, we say that $\alpha d_x(x,y)$ *and* $\beta d_y(x,y)$ *are alternating descent ascent directions.* □

This formality includes many of the most popular algorithms minmax algorithms. Here are some examples:

1. Gradient Descent Ascent: $d_x(x, y) = -\alpha \boldsymbol{\nabla}_x f(x, y)$ and $d_y(x, y) = \beta \boldsymbol{\nabla}_y f(x, y)$ with α and $\beta \in (0, +\infty)$

2. Gradient Descent multiple Ascent: $d_x(x, y) = -\alpha \boldsymbol{\nabla}_x f(x, y)$ and $d_y(x, y) = \sum_{k=1}^{n} \boldsymbol{\nabla}_y f(x, \tilde{y}_k)$, with α and $\beta \in (0, +\infty)$ and where $\boldsymbol{\nabla}_y f(x, \tilde{y}_k)$ is implicitly defined by

$$\begin{aligned} \tilde{y}_1 &= y + \beta \boldsymbol{\nabla}_y f(x, y) \\ \tilde{y}_2 &= \tilde{y}_1 + \beta \boldsymbol{\nabla}_y f(x, \tilde{y}_1) \\ &\vdots \end{aligned}$$

3. GradaMax: $d_x(x, y) = -\alpha \boldsymbol{\nabla}_x f(x, y)$ with $\alpha \in (0, +\infty)$ and $d_y(x, y) \in \arg\max_{d_y : y + d_y \in \mathcal{Y}} f(x, y + d_y)$, $\beta = 1$.

4. Alternating minmax: $d_x(x, y) \in \arg\min_{d_x : x + d_x \in \mathcal{X}} f(x + d_x, y)$ and $d_y(x, y) \in \arg\max_{d_y : y + d_y \in \mathcal{Y}} f(x, y + d_y)$.

Other methods popular in the robust training community such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) can also be expressed as alternating directions minmax.

A notable characteristic of alternating descent ascent sequences is that each player takes an action without taking into consideration what will be the consequences on the other player's action. Instead, we argue in this chapter for an approach where $d_x(x, y)$ and $d_y(x, y)$ are computed simultaneously, each player choosing their action while taking into account the other player's move. This is captured in the following definition.

Definition 4.1.5 (Full descent ascent) *We say that the sequence defined by* (4.2) *is a full descent ascent sequence if it satisfies*

$$f(x^{k+1}, y^{k+1}) \leqslant f(x^k, y^k + \beta^k d_y(x^k, y^k)) \tag{4.4a}$$

$$f(x^{k+1}, y^{k+1}) \geqslant f(x^{k+1}, y^k). \tag{4.4b}$$

with at least one of the inequalities holding strictly. By extension, we say that $\alpha d_x(x, y)$ *and* $\beta d_y(x, y)$ *are full descent ascent directions.* □

Fundamentally, the full descent ascent captures the nature of minmax optimizations. Not only the descent ascent step choices are, by construction, asymmetric, but it also reflects the fact the minimization needs to chose their step considering what will be the action of the max.

Remark 4.1.6 (Solving minmax as a full descent ascent algorithm) *If one uses the GradMax (as defined above), then the sequence* (4.2) *could asymptotically converge towards a local minmax, most notable if* $f(\cdot)$ *is strongly convex in* x *and strongly concave in* y*. Now, consider an analogous choice of full descent ascent directions given by:*

$$d_x(x^k, y^k) \in \underset{d_x : d_x + x^k \in \mathcal{X}}{\arg\min}\ f(x + d_x, y + d_y(x + d_x, y))$$

$$d_y(x^{k+1}, y^k) = \underset{d_y : y + d_y \in \mathcal{Y}}{\arg\max} f(x^{k+1}, y^k + d_y).$$

where we assume the arg max *is uniquely achieved. This choice of directions is **exactly** the solution of the minmax optimization. Evidently, one does not have access to closed form expressions of such functions, as this is the goal itself of an optimization algorithm. However, this shows how the full descent ascent directions describe a more appropriate concept of direction to find mimnax points.*

4.2 Obtaining local full descent ascent directions

In order to obtain local $d_x(x, y)$ and $d_y(x, y)$, it is usefull to consider the following result from minimization. Suppose one wants to solve the problem $\min_{x\in\mathcal{X}} f(x)$ where $\mathcal{X}$ is a convex set. If $f(\cdot)$ is continuously differentiable, projected direction methods solve this optimization by generating a sequence $x^{k+1} = x^k + \alpha d_x(x^k)$ where $d_x(x)$ is a local descent direction obtained from solving the quadratic subproblem

$$d_x(x) = \underset{d_x : d_x + x \in \mathcal{X}}{\arg\min}\, f(x) + d_x' \boldsymbol{\nabla}_x f(x) + \frac{1}{2} d_x' A(x) d_x \tag{4.5}$$

where $A(x)$ is a strictly positive definite matrix and $\alpha \in (0, 1]$. A large number of optimization methods can be written in this form including gradient descent (choosing $A(x)$ as the identity matrix), Newton method (choosing $A(x)$ as the Hessian matrix), Gauss-Newton method and its generalizations, Quasi-Newton methods, Trust Region methods (by also including a constraint on the norm of d_x) among many others.

In an analogous way, if $f(x, y)$ is differentiable in y, we define $d_y(x, y)$ as the solution of

$$d_y(x, y) = \underset{d_y : y + d_y \in \mathcal{Y}}{\arg\max} f(x, y) + d_y' \boldsymbol{\nabla}_y f(x, y) - \frac{1}{2} d_y' B(x, y) d_y \tag{4.6}$$

where $B(x, y)$ is a positive definite matrix. It is important to emphasize that $d_y(x, y)$ is function both of x and y. Consider the function $\hat{f}_x(x, y)$ defined by

$$\hat{f}_x(x, y) := f(x, y + \beta d_y(x, y)). \tag{4.7}$$

If $\hat{f}_x(x, y)$ is differentiable in x, we define $d_x(x, y)$ by

$$d_x(x, y) = \underset{d_x : x + d_x \in \mathcal{X}}{\arg\min} \hat{f}_x(x, y) + d_x' \boldsymbol{\nabla}_x \hat{f}_x(x, y) + \frac{1}{2} d_x' A(x, y) d_x \tag{4.8}$$

where $A(x, y)$ is a positive definite matrix. We can now state our first result

Proposition 4.2.1 (Computing local directions) *If $f(x, y)$ is continuously differentiable with respect to y and $\hat{f}_x(x, y)$ is continuously differentiable with respect to x on a* *neighborhood around a point* $(\tilde{x}, \tilde{y})$ *which is not a stationary point, then there exist* α_0 *and* β_0 *such that* $\forall \alpha \in (0, \alpha_0)$ *and* $\forall \beta \in (0, \beta_0)$ $\alpha d_x(\tilde{x}, \tilde{y})$ *and* $\beta d_y(\tilde{x}, \tilde{y})$ *are full descent ascent directions.* □

Proof. Consider the equations

$$f(x^{k+1}, y^{k+1}) = \hat{f}(x^k, y^k) + \alpha d_x(x^k, y^k)' \boldsymbol{\nabla}_x \hat{f}(x^k, y^k) + o(\alpha)$$
$$f(x^{k+1}, y^{k+1}) = f(x^{k+1}, y^k) + \beta d_y(x^{k+1}, y^k)' \boldsymbol{\nabla}_y f(x^{k+1}, y^k) + o(\beta)$$

where we use the fact that $f(x^{k+1}, y^{k+1}) = \hat{f}(x^{k+1}, y^k)$. As the functions are continuously differentiable, there exist α_0 and β_0 such that $\forall \alpha \in (0, \alpha_0)$ and $\forall \beta \in (0, \beta_0)$ the terms $o(\alpha)$ and $o(\beta)$ are dominated. From (4.6) and (4.8), we have that $d_y(x^k, y^k)' \boldsymbol{\nabla}_y f(x^k, y^k) \geqslant 0$ and $d_x(x^k, y^k)' \boldsymbol{\nabla}_x \hat{f}(x^k, y^k) \leqslant 0$. As at least either $d_y(x^k, y^k)' \boldsymbol{\nabla}_y f(x^k, y^k)$ or $d_x(x^k, y^k)' \boldsymbol{\nabla}_x \hat{f}(x^k, y^k)$ is non zero, otherwise (x^k, y^k) would be a stationary point, then they are full descent ascent directions. ■

We will now look at two particular choices of matrices $A(x, y)$ and $B(x, y)$ that will also help understanding the algorithm. In both we will consider the unconstrained case ($\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} = \mathbb{R}^m$).

4.2.1 Full LOLA

The first case is when one chooses $A(x, y)$ and $B(x, y)$ as the identity matrix, which is what we call the full LOLA. The direction for the max is

$$\beta d_y(x, y) = \beta \boldsymbol{\nabla}_y f(x, y).$$

For the min, the direction is

$$d_x(x, y) = -\boldsymbol{\nabla}_x f(x, y + \beta \boldsymbol{\nabla}_y f(x, y))$$

$$= -\,\partial_x f(x, y + \beta \boldsymbol{\nabla}_y f(x,y)) - \beta \boldsymbol{\nabla}_{xy} f(x,y) \partial_y f(x, y + \beta \boldsymbol{\nabla}_y f(x,y)).$$

If one linearizes this direction around (x, y) one obtains

$$d_x(x,y) = -\boldsymbol{\nabla}_x f(x,y) - \beta \boldsymbol{\nabla}_{xy} f(x,y) \boldsymbol{\nabla}_y f(x,y)$$

i.e. the standard LOLA direction.

Using these results, the full descent ascent sequence is

$$\begin{aligned} x^{k+1} &= x^k - \alpha^k \boldsymbol{\nabla}_x f(x^k, y^k + \beta^k \boldsymbol{\nabla}_y f(x^k, y^k)) \\ y^{k+1} &= y^k + \beta^k \boldsymbol{\nabla}_y f(x^{k+1}, y^k). \end{aligned}$$

In contrast with the standard (alternating) gradient descent ascent, in (full) LOLA, the descent direction uses the gradient of the maximzer to correct the direction towards where it should go. In the case where case where (x^k, y^k) is a local maximum, both the full and standard LOLA are equivalent to a gradient descent ascent as $\boldsymbol{\nabla}_y f(x^k, y^k) = 0$.

4.2.2 Full Newton types algorithms

Full descent ascent algorithms can also be used as Newton types algorithms by choosing matrices $A(x, y)$ and $B(x, y)$ as Hessian. For the maximizer, the straightforward choice of matrix is $B(x, y) = -\boldsymbol{\nabla}_{yy} f(x, y)$. For the minimizer there are two options. The first option is to take $A(x, y) = \boldsymbol{\nabla}_{xx} f(x, y)$ and the secondis to take

$$A(x,y) = \boldsymbol{\nabla}_{xx} f\Big(x, y - \beta \boldsymbol{\nabla}_{yy} f(x,y)^{-1} \boldsymbol{\nabla}_y f(x,y)\Big)$$

Taking $A(x, y) = \boldsymbol{\nabla}_{xx} f(x, y)$ has the advantage of making the differentiation easier, while in the second option we more closely maintain the spirit of full descent ascent of minimizing the cost of the future direction.

Remark 4.2.2 (Differentiability of $\hat{f}_x(x, y)$) *The assumption of differentiability of $\hat{f}_x(x, y)$ with respect to x is closely related to the differentiability of $d_y(x, y)$, which is* *known as sensitivity analysis. In the case where (x, y) is an interior point of the constrain set (or, equivalently, if $\mathcal{Y} = \mathbb{R}^m$) a sufficient condition is that $\boldsymbol{\nabla}_y f(x, y)$ and $B(x, y)$ are differentiable. However, establishing differentiability in the case where (x, y) is not an interior point is substantially more challenging, and naming such conditions goes beyond the scope of this chapter. We refer the reader to [121] which has a thorough treatment of the topic.* □

Remark 4.2.3 (Using momentum) *In minimization, algorithms with momentum are of the general form $x^{k+1} = x^k - \alpha(\boldsymbol{\nabla}_x f(x^k) + p_x^k)$. One example of such algorithm is to use $p_x^k = \boldsymbol{\nabla}_x f(x^{k-1}) + \mu\, p_x^{k-1}$ with $\mu \in [0, 1]$.*

The framework of full descent ascent also allows for methods with momentum by substituting $\boldsymbol{\nabla}_y f_x(x, y)$ by $\boldsymbol{\nabla}_y f_x(x, y) + p_y$ in (4.6) and $\boldsymbol{\nabla}_x \hat{f}_x(x, y)$ by $\boldsymbol{\nabla}_x \hat{f}_x(x, y) + p_x$ in (4.8), although these might no longer be full descent ascent directions as we define in Definition 4.1.5. □

4.3 Asymptotic convergence

Our goal now is to obtain conditions such that every limit point of the sequence

$$x^{k+1} = x^k + \alpha^k d_x(x^k, y^k)$$
$$y^{k+1} = y^k + \beta^k d_y(x^{k+1}, y^k)$$

where $d_x(x^k, y^k)$ is given by (4.8) and $d_y(x^{k+1}, y^k)$ is given by (4.6) is a stationary point. We will not make any assumption of convexity or concavity instead casting the results in the most general possible way. For this reason, our convergence results will pertain to asymptotic properties of full descent ascent sequences. Results on non asymptotic convergence will be the subject of a future work.

For the sake of conciseness, we will state our using the notation

$$\hat{f}_x(x, y) = f(x, y + \beta d_y(x, y)).$$

Let us denote by $\lambda_{min}(M)$ and $\lambda_{max}(M)$ the smallest and largest eigenvalues of a symmetric matrix M. In addition to the assumptions of convexity and closeness of $\mathcal{X}$ and $\mathcal{Y}$ and the continuous differentiability of $f(\cdot)$ and $\hat{f}_x(\cdot)$ we will also need the following assumptions.

Assumption 4.3.1 *Given a full descent ascent sequence $\{(x^k, y^k)\}$, for all k the eigenvalues of $A(x^k, y^k)$ and $B(x^k, y^k)$ are bounded by bellow and above and away from zero, meaning that there exist positive constants c_1, c_2, c_3, c_4 such that $\forall k > 0$,*

$$\lambda_{min}(A(x^k, y^k)) > c_1 \qquad \lambda_{max}(A(x^k, y^k)) < c_2$$
$$\lambda_{min}(B(x^k, y^k)) > c_3 \qquad \lambda_{max}(B(x^k, y^k)) < c_4 \qquad \square$$

This assumption essentially guarantees that optimizations (4.6) and (4.8) will always be well defined and only have one solution. It is important to emphasize that $A(x, y)$ and $B(x, y)$ are algorithmic choices in the sense that they are chosen by the practitioner.

Our first result concerns the convergence when the matrices $A(x^k, x^k)$ and $B(x^k, x^k)$ and the step sizes α^k, β^k are constant.

Theorem 4.3.2 (Constant step size) *Let $\{(x^k, y^k)\}$ be a full descent ascent sequence with $\alpha^k = \beta^k = 1$, $A(x, y) = A$ and $B(x, y) = B$. Assume that $\forall x_1, x_2 \in \mathcal{X}$ and $\forall y_1, y_2 \in \mathcal{Y}$ there exist constants $L_x, L_y > 0$ such that the following smoothness condition holds:*

$$\|\nabla_x f(x_1, y_1) - \nabla_x f(x_2, y_2)\| < L_x \sqrt{\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2}$$
$$\|\nabla_y f(x_1, y_1) - \nabla_y f(x_2, y_2)\| < L_y \sqrt{\|x_1 - x_2\|^2 + \|y_1 - y_2\|^2}$$

If $2\Big(L_x\sqrt{1+\lambda_{min}(B)\,L_y^2}\Big)^{-1} > \lambda_{min}(A)^{-1}$ *and* $2\,L_y^{-1} > \lambda_{min}(B)^{-1}$ *then every limit point of* $\{(x^k, y^k)\}$ *is a stationary point of* $f(\cdot)$. *Moreover,* $d_x(x,y)$ *and* $d_y(x,y)$ *are full descent ascent directions, meaning that*

$$\begin{aligned} f(x^{k+1}, y^{k+1}) &\leqslant f(x^k, y^k + d_y(x^k, y^k)) \\ f(x^{k+1}, y^{k+1}) &\geqslant f(x^{k+1}, y^k) \end{aligned}$$

with at least one of the inequalities holding strictly. □

In order to prove this theorem, we need the results from the following Lemma:

Lemma 4.3.3 (Simultaneous descent ascent are gradient related) *The full descent ascent directions* $d_x(x,y)$ *and* $d_y(x,y)$ *are gradient related meaning that for any sequence* $\{(x^k, y^k)\}$ *that converges to a nonstationary point, then the corresponding sequence* $\{(d_x(x^k, y^k), d_y(x^k, y^k))\}$ *is bounded and satisfies*

$$\begin{aligned} \limsup_{k\to\infty} d_x(x^k, y^k)\,'\boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) &\leqslant 0 \\ \liminf_{k\to\infty} d_y(x^{k+1}, y^k)\,'\boldsymbol{\nabla}_y f(x^{k+1}, y^k) &\geqslant 0 \end{aligned}$$ □

with at least one inequality holding strictly and where $\hat{f}_x(x,y)$ *is defined in* (4.7).

Proof. The proof is inspired in the proof of Prop 3.3.1 of [122].

Assume that $\{(x^k, y^k)\}$ converges to a non stationary point $(\tilde{x}, \tilde{y})$. We need to prove the following four equations

$$\limsup_{k\to\infty} \left\| d_x(x^k, y^k) \right\| < \infty \tag{4.9a}$$

$$\limsup_{k\to\infty} \left\| d_y(x^k, y^k) \right\| < \infty \tag{4.9b}$$

$$\liminf_{k\to\infty} d_y(x^{k+1}, y^k)\,'\boldsymbol{\nabla}_y f(x^{k+1}, y^k) \geqslant 0 \tag{4.9c}$$

$$\limsup_{k\to\infty} d_x(x^k, y^k)\,'\boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) \leqslant 0 \tag{4.9d}$$

By continuity of the projection (see Prop. 1.1.4 in [122]) and the differential continuity of $\boldsymbol{\nabla}_x \hat{f}_x(x,y)$ and $\boldsymbol{\nabla}_y f(x,y)$

$$\lim_{k\to\infty} \left\| d_y(x^k,y^k) \right\| = \|d_y(\tilde{x},\tilde{y})\| < \infty$$

$$\lim_{k\to\infty} \left\| d_x(x^k,y^k) \right\| = \|d_x(\tilde{x},\tilde{y})\| < \infty$$

which proves (4.9b) and (4.9a). To prove (4.9c) and (4.9d), first remember the property that, for any continuously differentiable function $\phi(x)$ on a convex set $\mathcal{X}$, if x^* is a local minimum, then $\boldsymbol{\nabla}\phi(x^*)'(x-x^*) \geqslant 0 \ \forall x \in \mathcal{X}$; there is an equivalent property for a local maximum. Applying these condition to (4.6) and (4.8) we obtain

$$\big(B(x^{k+1},y^k)d_y(x^{k+1},y^k) - \boldsymbol{\nabla}_y f(x^{k+1},y^k)\big)'(\tilde{d}_y - d_y(x^{k+1},y^k)) \geqslant 0 \quad \forall \tilde{d}_y : d_y + y^k \in \mathcal{Y}$$

$$\Big(A(x^k,y^k)d_x(x^k,y^k) + \boldsymbol{\nabla}_x \hat{f}_x(x^k,y^k)\Big)'(\tilde{d}_x - d_x(x^k,y^k)) \geqslant 0 \quad \forall \tilde{d}_x : d_x + x^k \in \mathcal{X}$$

The above equations hold for $(\tilde{d}_x, \tilde{d}_y) = (0,0)$ which yields

$$\boldsymbol{\nabla}_x \hat{f}_x(x^k,y^k)'d_x(x^k,y^k) \leqslant -d_x(x^k,y^k)A(x^k,y^k)d_x(x^k,y^k) \leqslant -c_1 \left\| d_x(x^k,y^k) \right\|^2 \tag{4.11a}$$

$$\boldsymbol{\nabla}_y f(x^{k+1},y^k)'d_y(x^{k+1},y^k) \geqslant d_y(x^{k+1},y^k)'B(x^{k+1},y^k)d_y(x^{k+1},y^k) \geqslant c_3 \left\| d_y(x^{k+1},y^k) \right\|^2 \tag{4.11b}$$

where the last inequality is taken from the boundness of the eigenvalues of $A(x^k,y^k)$ and $B(x^{k+1},y^k)$. Taking the limit we obtain

$$\liminf_{k\to\infty} d_y(x^{k+1},y^k)\,'\boldsymbol{\nabla}_y f(x^{k+1},y^k) \geqslant c_3 \|d_y(\tilde{x},\tilde{y})\|^2 \geqslant 0$$

$$\limsup_{k\to\infty} d_x(x^k,y^k)\,'\boldsymbol{\nabla}_x \hat{f}_x(x^k,y^k) \leqslant -c_1 \|d_x(\tilde{x},\tilde{y})\|^2 \leqslant 0$$

with at least one inequality holding strictly because $(\tilde{x},\tilde{y})$ is not a stationary point. ■

Proof of Theorem 4.3.2. Let us start proving the property for the max. Using the property known as the ascent lemma for Lipschitz function (see Prop. A.24 in [122]) we have that

$$f(x^{k+1}, y^k + d_y(x^k, y^k)) - f(x^{k+1}, y^k) \geqslant \boldsymbol{\nabla}_y f(x^{k+1}, y^k)' d_y(x^k, y^k) - \frac{L_y}{2}\left\| d_y(x^{k+1}, y^k) \right\|$$

Combining this result with (4.11b) where $c_3 := \lambda_{min}(B)$ we obtain

$$f(x^{k+1}, y^{k+1}) - f(x^{k+1}, y^k) \geqslant \left(\lambda_{min}(B) - \frac{L_y}{2}\right)\left\| d_y(x^{k+1}, y^k) \right\| \geqslant 0$$

where the right most inequalities hold because $\lambda_{min}(B) > L_y/2$. So if $(\bar{x}, \bar{y})$ is a limit point of a subsequences $\{(x^k, y^k)\}_{\mathcal{K}}$ then

$$\lim_{k\to\infty, k\in\mathcal{K}} f(x^{k+1}, y^{k+1}) - f(x^{k+1}, y^k) = 0$$

implying, by continuity of the projection, that $\|d_y(\bar{x}, \bar{y})\| = 0$.

For the min, take $\hat{f}_x(x, y)$ as defined in (4.7) and consider the following inequalities

$$\begin{aligned}
&\left\| \boldsymbol{\nabla}_x \hat{f}_x(x + d_x(x, y), y) - \boldsymbol{\nabla}_x \hat{f}_x(x, y) \right\| \\
&= \left\| \boldsymbol{\nabla}_x f\Big(x + d_x(x, y), y + d_y\Big(x + d_x(x, y), y\Big)\Big) - \boldsymbol{\nabla}_x f(x, y + d_y(x, y)) \right\| \\
&\leqslant L_x \sqrt{\|d_x(x, y)\|^2 + \left\| d_y\Big(x + d_x(x, y), y\Big) - d_y(x, y) \right\|^2} \\
&\leqslant L_x \sqrt{\|d_x(x, y)\|^2 + \left\| B^{-1} \boldsymbol{\nabla}_y f\Big(x + d_x(x, y), y\Big) - B^{-1} \boldsymbol{\nabla}_y f(x, y) \right\|^2} \\
&\leqslant L_x \sqrt{\|d_x(x, y)\|^2 + \lambda_{min}(B) \left\| \boldsymbol{\nabla}_y f\Big(x + d_x(x, y), y\Big) - \boldsymbol{\nabla}_y f(x, y) \right\|^2} \\
&\leqslant L_x \sqrt{\|d_x(x, y)\|^2 + \lambda_{min}(B)\, L_y^2 \|d_x(x, y)\|^2} \\
&= L_x \sqrt{1 + \lambda_{min}(B)\, L_y^2} \|d_x(x, y)\|
\end{aligned}$$

where in the third line we used the fact that projections are nonexpansive (see Prop. 1.1.4 in [122]). These imply that the function $\hat{f}_x(x, y)$ is also smooth with constant $L_x\sqrt{1 + \lambda_{min}(B)\, L_y^2}$. So using the equivalent steps as for the max we arrive to

$$\hat{f}_x(x^k + d_x(x^k, y^k), y^k) - \hat{f}_x(x^k, y^k) \leqslant \left(\frac{L_x\sqrt{1 + \lambda_{min}(B)\, L_y^2}}{2} - \lambda_{min}(A) \right) \left\| d_x(x^k, y^k) \right\| \leqslant 0$$

where the right most equality hold because $\lambda_{min}(A) > L_x\sqrt{1 + \lambda_{min}(B)\, L_y^2}\ /2$. So if $(\bar{x}, \bar{y})$ is a limit point of a subsequences $\{(x^k, y^k)\}_{\mathcal{K}}$ then

$$\lim_{k\to\infty, k\in\mathcal{K}} \hat{f}_x(x^{k+1}, y^k) - \hat{f}_x(x^k, y^k) = 0$$

implying, by continuity of the projection, that $\|d_x(\bar{x}, \bar{y})\| = 0$. Therefore that $(\bar{x}, \bar{y})$ is a stationary point. ■

It is easier to interpret Theorem 4.3.2 when $\mathcal{X} = \mathbb{R}^n$, $\mathcal{Y} = \mathbb{R}^m$ The full descent ascent directions are

$$\begin{aligned} d_y(x, y) =& B^{-1}\nabla_y f(x, y) \\ d_x(x, y) =& -A^{-1}\nabla_x f(x, y + B^{-1}\nabla_y f(x, y)). \end{aligned}$$

Now if we use the fact that $\lambda_{max}(A^{-1}) = \lambda_{min}(A)^{-1}$ and equivalent to B, Theorem 4.3.2 essentially says two things. The first one is that the larger the constants L_x, L_y are, the smaller the step sizes, represented by $\lambda_{min}(A)^{-1}$ and $\lambda_{min}(B)^{-1}$, can be. This kind of result is typical in optimization. But the second particularly interesting thing is that the maximum step size of the minimizer depends on the step size of maximizer, essentially stating that if the maximizer take small steps, the minimizer also needs to take small steps. The idea that the minimizer needs to take smaller steps than the maximizer is common in minmax optimization (see for instance the discussion for Gradient Descent Ascent on [32]). What is innovative in our result is that we are able to quantify exactly how big the step can be.

The biggest limitation of Theorem 4.3.2 is that one often does not know the values of L_x and L_y. As a consequence, one would need to manually tune the matrices A and B using a on trial and error, and the step sizes are rarely as large as they could be. Our next result uses an Armijo type condition and a backtracking algorithm to determine the step sizes. We point out that the result does not require the smoothness condition.

Take two scalars $\sigma_x, \sigma_y \in (0,1)$. At a given point (x^k, y^k), given two step sizes (α^k, β^k), we define the following Armijo type conditions for the minmax

$$f(x^{k+1}, y^{k+1}) - \hat{f}_x(x^k, y^k) \leqslant \sigma_x \, \alpha^k \, d_x(x^k, y^k)' \boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) \tag{4.12a}$$

$$f(x^{k+1}, y^{k+1}) - f(x^{k+1}, y^k) \geqslant \sigma_y \, \beta^k \, d_y(x^{k+1}, y^k)' \boldsymbol{\nabla}_y f(x^k, y^k) \tag{4.12b}$$

with at least one of the inequalities holding strictly and where $\hat{f}_x(x, y)$ is given by (4.7). We bring attention to the reader that $d_x(x, y)$ and $\hat{f}(x, y)$ depend on the value of β^k. These Armijo conditions not only guarantee that $\alpha^k d_x(x^k, y^k)$ and $\beta^k d_y(x^{k+1}, y^k)$ are full descent ascent directions, but also guarantees that at each iteration the steps are sufficiently large. We use these conditions to design Algorithm 2 and prove its convergence.

Algorithm 2 Simultaneous descent ascent with Armijo rule

```
Require: An initial point $(x^0, y^0)$ and rates $r_x, r_y \in (0,1)$
 1: $(\alpha^k, \beta^k) = (1,1)$
 2: $F_{min} = \hat{f}_x(x^k, y^k) + \sigma_x \alpha^k d_x(x^k, y^k)' \nabla_x \hat{f}(x^k, y^k)$
 3: $F_{max} = f(\tilde{x}, y^k) + \sigma_y \beta^k d_y(\tilde{x}, y^k)' \nabla_y f(\tilde{x}, y^k)$
 4: while $f(\tilde{x}, \tilde{y}) - F_{min} > 0$ and $f(\tilde{x}, \tilde{y}) - F_{max} < 0$ do
 5:     $\tilde{x} = x^k + \alpha^k d_x(x^k, y^k)$
 6:     $\tilde{y} = y^k + \beta^k d_y(\tilde{x}, y^k)$
 7:     $F_{min} = \hat{f}_x(x^k, y^k) + \sigma_x \alpha^k d_x(x^k, y^k)' \nabla_x \hat{f}(x^k, y^k)$
 8:     $F_{max} = f(\tilde{x}, y^k) + \sigma_y \beta^k d_y(\tilde{x}, y^k)' \nabla_y f(\tilde{x}, y^k)$
 9:     if $f(\tilde{x}, \tilde{y}) - F_{min} > 0$ then
10:         $\alpha^k = \alpha^k r_x$
11:     end if
12:     if $f(\tilde{x}, \tilde{y}) - F_{max} < 0$ then
13:         $\beta^k = \beta^k r_y$
14:     end if
15: end while
16: $x^{k+1} = \tilde{x}$
17: $y^{k+1} = \tilde{y}$
18: $k = k + 1$
19: Go to 1
```

Theorem 4.3.4 (Convergence of Armijo) *Every limit point of a sequence $\{(x^k, y^k)\}$ generated by Algorithm 2 is a stationary point.* □

Proof. This proof is inspired by the proof of Prop. 1.2.1 in [122]. Take $\hat{f}_x(x, y)$ as defined in (4.7) and, in order to have shorter expressions, let us define $d_x^k := d_x(x^k, y^k)$

and $d_y^k := d_y(x^{k+1}, y^k)$.

As $f(\cdot)$ and $\hat{f}_x(\cdot)$ are continuous function, then as $(\bar{x}, \bar{y})$ is a limit point of $\{(x^k, y^k)\}$ then $f(\bar{x}, \bar{y})$ is a limit point of $\{f(x^k, y^k)\}$ and equivalent to $\hat{f}_x(\cdot)$. Moreover, $f(\bar{x}, \bar{y})$ is also a limit point of $\{f(x^{k+1}, y^k)\}$.

Starting with the max. From the previous argument, we have that

$$f(x^{k+1}, y^k) - f(x^{k+1}, y^{k+1}) \to 0.$$

By the choice of direction in (4.8) we have that $d_y^k{}' \boldsymbol{\nabla}_y f(x^{k+1}, y^k) \geqslant 0$. Combining this with the Armijo rule in (4.12b) we have that

$$f(x^{k+1}, y^k) - f(x^{k+1}, y^{k+1}) \leqslant -\sigma_y \beta^k d_y^k{}' \boldsymbol{\nabla}_y f(x^{k+1}, y^k) \leqslant 0 \tag{4.13}$$

Therefore we obtain that

$$\lim_{k\to\infty} \beta^k d_y^k{}' \boldsymbol{\nabla}_y f(x^{k+1}, y^k) = 0 \tag{4.14}$$

Now the min. Combining (4.11b) and (4.14) implies that $\beta^k \, d_y^k \to \mathbf{0}_n$. And as $\hat{f}_x(\cdot)$ is continuous, we obtain

$$\hat{f}_x(x^k, y^k) - f(x^{k+1}, y^{k+1}) \to 0.$$

By the choice of descent direction we have that $d_x^k{}' \boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) \leqslant 0$, and by the Armijo rule

$$\hat{f}_x(x^k, y^k) - f(x^{k+1}, y^{k+1}) \geqslant -\sigma_x \, \alpha^k d_x^k{}' \boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) \geqslant 0$$

Therefore we obtain

$$\lim_{k\to\infty} \alpha^k d_x^k{}' \boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k) = 0 \tag{4.15}$$

As d_x^k and d_y^k are gradient related from Lemma 4.3.3, in order for (4.14) and (4.15) to hold simultaneously either $(\bar{x}, \bar{y})$ is a stationary point or $\alpha^k \to 0$ or $\beta^k \to 0$.

We will assume, in order to arrive to a contradiction, that $(\bar{x}, \bar{y})$ is not a stationary. We will start by assuming that $\{\beta^k\} \to 0$, and show that it implies that $\{\alpha^k\} \to 0$ and then show it leads to a contradiction.

The core argument used to prove the contradiction relies in the following observation. If $\{\beta^k\} \to 0$ it means that there exist a $\bar{k}$, such that for each $k > \bar{k}$

$$f\big(x^k + \alpha^k d_x^k, y^k\big) - f\left(x^k + \alpha^k d_x^k, y^k + \frac{\beta}{r_y}^k d_y^k\right) > -\frac{\beta}{r_y}^k \sigma_y d_y^k{}' \boldsymbol{\nabla}_y f\big(x^k + \alpha^k d_x^k, y^k\big). \quad (4.16)$$

This equation holds because $\{\beta^k\} \to 0$ implies that the alternating backtracking algorithm will always need to run at least one time after some point, which we called $\bar{k}$. If the alternating backtracking algorithm ran at least one time it means that the Armijo conditions for the max was not verified for β^k/r_y, otherwise there would not have been the need to run another iteration of the backtracking, which justifies (4.16).

Since the search direction d_y^k is gradient related, then $\{d_y^k\}$ is bounded and so there exists a subsequences $\{d_y^k\}_{\bar{\mathcal{K}}}$ of $\{d_y^k\}$ such that $\{d_y^k\}_{\bar{\mathcal{K}}}$ converges to some point $\bar{d}_y$. Then, $\forall k \in \bar{\mathcal{K}}, k > \bar{k}$

$$\frac{f(x^k + \alpha^k d_x^k, y^k) - f(x^k + \alpha^k d_x^k, y^k + \beta^k/r_y d_y^k)}{\beta^k/r_y} > -\sigma_y d_y^k{}' \boldsymbol{\nabla}_y f(x^k + \alpha^k d_x^k, y^k) \quad (4.17)$$

By the mean value theorem, this relation can be written as

$$-d_y^k{}' \boldsymbol{\nabla}_y f(x^k + \alpha^k d_x^k, y^k + \tilde{\beta}^k d_y^k) > -\sigma_y d_y^k{}' \boldsymbol{\nabla}_y f(x^k + \alpha^k d_x^k, y^k) \quad (4.18)$$

with $\tilde{\beta}^k \in [0, \beta^k/r_y]$. Now taking the limit as $k \to \infty, k \in \bar{\mathcal{K}}$ and because $\{\beta^k\} \to 0$ we obtain

$$-\bar{d}_y{}' \boldsymbol{\nabla}_y f(\bar{x}, \bar{y}) \geqslant -\sigma_y \bar{d}_y{}' \boldsymbol{\nabla}_y f(\bar{x}, \bar{y}) \Leftrightarrow 0 \geqslant (1 - \sigma_y) \bar{d}_y{}' \boldsymbol{\nabla}_y f(\bar{x}, \bar{y}) \Rightarrow 0 \geqslant \bar{d}_y{}' \boldsymbol{\nabla}_y f(\bar{x}, \bar{y}).$$

There are two possible cases. The first one is that the last inequality holds strictly, *i.e.* $\bar{d_y}'\nabla_y f(\bar{x},\bar{y}) < 0$. This contradicts the assumption that $\bar{d_y}$ is gradient related, therefore this case is not possible. The second case is that $\bar{d_y}'\nabla_y f(\bar{x},\bar{y}) = 0$. By contradiction assumption, $(\bar{x},\bar{y})$ is not a stationary point, meaning $\lim_{k\to\infty} \alpha^k d_x^k{}'\nabla_x \hat{f}_x((x^k,y^k) \neq 0$ (otherwise $(\bar{x},\bar{y})$ is a stationary point). By (4.15) this implies that $\{\alpha^k\} \to 0$.

Analogously to the previous case, if $\{\alpha^k\} \to 0$ then there exist a $\bar{k}$ such that for each $k > \bar{k}$

$$\hat{f}_x(x^k,y^k) - \hat{f}_x\left(x^k + \frac{\alpha^k}{r_x}d_x^k, y^k\right) < -\sigma_x \frac{{\alpha^k}^k}{r_x} d_x^k{}'\nabla_x \hat{f}_x(x^k,y^k). \tag{4.19}$$

Using equivalent arguments as above, we arrive to the conclusion that there is a subsequences $\{d_x^k\}_{\bar{\mathcal{K}}}$ that converges to some point $\bar{d_x}$ and that satisfies $0 \leqslant \bar{d_x}'\nabla_x f(\bar{x},\bar{y})$ which, if the inequality is strict, contradicts the assumption that d_x is gradient related , or contradicts the proof assumption that $(\bar{x},\bar{y})$ is not a stationary point. Therefore, by contradiction $(\bar{x},\bar{y})$ is a stationary point. ■

The idea behind Algorithm 2 is to obtain steps sizes α^k, β^k that satisfy the Armijo conditions by implementing a backtracking algorithm. A fundamental aspect of the algorithm is that α^k and β^k are updated only when they do not satisfy their respective Armijo conditions; this plays a crucial role in the proof of Theorem 4.3.4.

Theorem 4.3.2 and Theorem 4.3.4 guarantee that every limit point of the generated full descent ascent sequence is a stationary point, but they do not guarantee that such limit points exist. This is guaranteed by the next result, the Capture Theorem. The Capture Theorem essentially says that, if (x^*y^*) is an isolated local minmax, if one element $(x^{\bar{k}}, y^{\bar{k}})$ of the full descent ascent passes close enough to it, then $\{(x^k,y^k)\}$ will converge towards (x^*,y^*).

Theorem 4.3.5 (Capture Theorem) *Let $\{(x^k,y^k)\}$ be a sequence generated by the full descent ascent direction method using either the Theorem 4.3.2 or Theorem 4.3.4. Let*

(x^*, y^*) be an isolated local minmax on a neighborhood where it is also the only stationary point. Then there exist a neighborhood $S_x \subset \mathcal{X}$ around x^* and a neighborhood $S_y \subset \mathcal{Y}$ around y^* such that if for some $\bar{k}$, $(x^{\bar{k}}, y^{\bar{k}}) \in S_x \times S_y$ then $\lim_{k,k>\bar{k}}(x^k, y^k) = (x^*, y^*)$. □

Proof. This proof is inspired by the proof of Prop. 1.2.4 of [122]. Let the interval $[0, \delta_0]$ and the function $h(\cdot)$ be the ones associated to the local minmax (x^*, y^*) according to Definition 4.1.1 . Take $\hat{f}_x(x, y)$ as defined in (4.7) and, in order to have shorter expressions, let us define $d_x^k := d_x(x^k, y^k)$ and $d_y^k := d_y(x^{k+1}, y^k)$.

Let us now take a specific $\delta \in [0, \delta_0]$. By definition, (x^*, y^*) is also a local minmax in that interval. Define for $t \in [0, \delta]$ and $t \in [0, h(\delta)]$ the functions

$$\phi_x(t, y) = \min_{x\in\mathcal{X}:t\leqslant\|x^*-x\|\leqslant\delta} \hat{f}_x(x, y) - \hat{f}_x(x^*, y^*)$$

$$\phi_y(t, x) = \max_{y\in\mathcal{Y}:t\leqslant\|y^*-y\|\leqslant h(\delta)} f(x, y) - f(x^*, y^*)$$

For a fixed y, $\phi_x(t, y)$ is an increasing function of t and for a fixed x, $\phi_y(t, x)$ is a decreasing function of t. Given any $\epsilon_x \in (0, \delta]$ and $\epsilon_y \in (0, h(\delta)]$, take $r_x \in (0, \epsilon_x]$ and $r_y \in (0, \epsilon_y]$ such that

$$\|x - x^*\| < r_x \quad \Rightarrow \quad \|x - x^*\| + c_1^{-1}\left\|\boldsymbol{\nabla}_x \hat{f}_x(x, y)\right\| < \epsilon_x \tag{4.20a}$$

$$\|y - y^*\| < r_y \quad \Rightarrow \quad \|y - y^*\| + c_3^{-1}\|\boldsymbol{\nabla}_y f(x, y)\| < \epsilon_y \tag{4.20b}$$

where c_1 and c_3 are from Assumption 4.3.1. Consider the open sets

$$S_x := \{x \in \mathcal{X} : \|x - x^*\| < \epsilon_x \text{ and } \forall y : \|y - y^*\| < \epsilon_y,\ \hat{f}_x(x, y) - f(x^*, y^*) < \phi_x(r_x, y)\}$$

$$S_y := \{y \in \mathcal{Y} : \|y - y^*\| < \epsilon_y \text{ and } \forall x : \|x - x^*\| < \epsilon_x,\ f(x, y) - f(x^*, y^*) < \phi_y(r_y, x)\}.$$

Now we prove that $x^k \in S_x \quad \Rightarrow \quad x^{k+1} \in S_x$ and that $y^k \in S_y \quad \Rightarrow \quad y^{k+1} \in S_y$. Starting with x^k, as $x^k \in S_x$ and $y^k \in S_y$, then

$$\phi_x\left(\left\|x^* - x^k\right\|, y^k\right) \leqslant \hat{f}_x(x^k, y^k) - f(x^*, y^*) < \phi_x(r_x, y^k)$$

where the right inequality derives from the definition of S_x and the left inequality from the definition of $\phi_x(\cdot)$. As $\phi_x(\cdot)$ is increasing in t, the previous relation implies $\left\|x^* - x^k\right\| < r_x$. Now we use the fact that in both Theorem 4.3.2 and 4.3.2 we have that $\alpha^k \leqslant 1$. Moreover, because the projection is a contracting map $\left\|d_x^k\right\| \leqslant c_1^{-1}\left\|\boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k)\right\|$ we obtain

$$\left\|x^{k+1} - x^*\right\| \leqslant \left\|x^k - x^*\right\| + \left\|\alpha^k d_x^k\right\| \leqslant \left\|x^k - x^*\right\| + c_1^{-1}\left\|\boldsymbol{\nabla}_x \hat{f}_x(x^k, y^k)\right\| \leqslant \epsilon_x$$

where the last inequality derives from (4.20a). Now looking back at the max from the previous equation we have that $\left\|x^{k+1} - x^*\right\| \leqslant \epsilon_x$ which implies

$$\phi_y\Big(\left\|y^* - y^k\right\|, x^{k+1}\Big) \geqslant f(x^{k+1}, y^k) - f(x^*, y^*) > \phi_y(r_y, x^{k+1}).$$

As $\phi_y(\cdot)$ is decreasing in t, the previous relation implies $\left\|y^* - y^k\right\| < r_y$. Now using the assumptions that $\beta^k \leqslant 1$ (same argument as α^k) and because the projection is a contracting map $\left\|d_y^k\right\| \leqslant c_3^{-1}\left\|\boldsymbol{\nabla}_x f(x^{k+1}, y^k)\right\|$ we obtain

$$\left\|y^{k+1} - y^*\right\| \leqslant \left\|y^k - y^*\right\| + \left\|\beta^k d_y^k\right\| \leqslant \left\|y^k - y^*\right\| + c_3^{-1}\left\|\boldsymbol{\nabla}_x f(x^{k+1}, y^k)\right\| \leqslant \epsilon_y$$

As $\{(x^k, y^k)\}$ is a full descent ascent sequence

$$\begin{cases} \hat{f}_x(x^{k+1}, y^k) - \hat{f}_x(x^*, y^*) \leqslant \hat{f}_x(x^k, y^k) - \hat{f}_x(x^*, y^*) < \phi_x(r_x, y^k) \\ \left\|x^{k+1} - x^*\right\| \leqslant \epsilon_x & \Rightarrow \quad x^{k+1} \in S_x \\ \left\|y^{k+1} - y^*\right\| \leqslant \epsilon_y \end{cases}$$

and

$$\begin{cases} f(x^{k+1}, y^{k+1}) - f(x^*, y^*) \leqslant f(x^{k+1}, y^k) - f(x^*, y^*) < \phi_y(r_y, x^{k+1}) \\ \left\|x^{k+1} - x^*\right\| \leqslant \epsilon_x & \Rightarrow \quad y^{k+1} \in S_y \\ \left\|y^{k+1} - y^*\right\| \leqslant \epsilon_y \end{cases}$$

Finally, by induction we have that if for some $\bar{k}$, $x^{\bar{k}} \in S_x$ and $y^{\bar{k}} \in S_y$, then $x^k \in S_x$ and $y^k \in S_y$ $\forall k > \bar{k}$. Let $\bar{S}_x$ and $\bar{S}_y$ be the closure of S_x and S_y. They are compact sets, therefore the sequence (x^k, y^k) must have at least one limit point which is a stationary point according to Theorem 4.3.2 and Theorem 4.3.4 . As the only stationary point is (x^*, y^*), therefore $(x^k, y^k) \to (x^*, y^*)$. ■

4.4 Conclusion

In this chapter, we have presented a new type of algorithm to solve minmax optimization using what we call full descent ascent directions. We have shown that such directions are better at generalizing the concept of descent direction from regular optimization. We were also able to state conditions that guarantee the asymptotic convergence of such algorithm to local minmax points.

While we have not found applications for which full descent ascent directions outperform the state of the art, they provide an elegant way to look at minmax optimization. Further exploration, both with respect to the theory and practice, could unfold cases in which such directions outperform other methods.

Bibliography

[1] R. Chinchilla, G. Yang, and J. P. Hespanha, "Newton and interior-point methods for (constrained) nonconvex-nonconcave minmax optimization with stability guarantees." Submitted for journal publication, May, 2022.

[2] R. Chinchilla and J. P. Hespanha, *Stochastic programming using expected value bounds*, *Transactions on Automatic Control.* Submitted Dec. 2020, Accepted June. 2022. To appear.

[3] J. P. Hespanha, R. Chinchilla, R. R. Costa, M. K. Erdal, and G. Yang, *Forecasting COVID-19 cases based on a parameter-varying stochastic SIR model*, *Annual Reviews in Control,* Pandemic Special Issue (2021).

[4] R. Chinchilla, G. Yang, M. K. Erdal, R. R. Costa, and J. P. Hespanha, *A Tale of Two Doses: Model Identification and Optimal Vaccination for COVID-19*, in *Proc. of the 60th IEEE Conf. on Decision and Contr.*, Dec., 2021.

[5] R. Chinchilla and J. P. Hespanha, *Optimization-based estimation of expected values with application to stochastic programming*, in *Proc. of the 58th IEEE Conf. on Decision and Contr.*, Dec., 2019.

[6] F. Orieux and R. Chinchilla, *Semi-unsupervised Bayesian convex image restoration with location mixture of Gaussian*, in *Proc. of the 25th IEEE European Signal Processing Conference*, Aug., 2017.

[7] F. Orieux and R. Chinchilla, *Restauration d'image par une approche bayesienne semi non supervisee et le melange de gaussienne*, in *26eme Colloque GRETSI Traitement du Signal et des Images, GRETSI 2017*, 2017.

[8] W. F. Costa, M. J. Bieleveld, R. G. Chinchilla, and A. M. Saraiva, *Segmentation of land use maps for precision agriculture*, in *Anais do VIII Workshop de Computacao Aplicada a Gestao do Meio Ambiente e Recursos Naturais*, SBC, 2017.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1).* Prentice Hall, 1993.

[10] J. C. Spall, *Introduction to stochastic search and optimization: estimation, simulation, and control*, vol. 65. John Wiley & Sons, 2005.

[11] A. C Atkinson, *Optimum Experimental Designs.* No. 8 in Oxford Statistical Science Series. Clarendon Press ; Oxford University Press, 1992.

[12] K. Chaloner and I. Verdinelli, *Bayesian experimental design: A review*, *Statistical Science* **10** (1995), no. 3 273–304.

[13] G. Monegato, *An overview of the computational aspects of kronrod quadrature rules*, *Numerical Algorithms* **26** (2001), no. 2 173–196.

[14] M. Evans and T. Swartz, *Approximating integrals via Monte Carlo and deterministic methods*, vol. 20, ch. 5. OUP Oxford, 2000.

[15] P. A. Samuelson, *The fundamental approximation theorem of portfolio analysis in terms of means, variances and higher moments*, in *Stochastic Optimization Models in Finance*, pp. 215–220. Elsevier, 1975.

[16] O. Loistl, *The erroneous approximation of expected utility by means of a taylor's series expansion: analytic and computational results*, *The American Economic Review* **66** (1976), no. 5 904–910.

[17] W. Hlawitschka, *The empirical nature of taylor-series approximations to expected utility*, *The American Economic Review* **84** (1994), no. 3 713–719.

[18] C. Robert and G. Casella, *Monte Carlo statistical methods.* Springer Science & Business Media, 2013.

[19] R. W. Butler, *Saddlepoint Approximations with Applications.* Cambridge University Press, 1 edition ed., 2007.

[20] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, *The sample average approximation method for stochastic discrete optimization*, *SIAM Journal on Optimization* **12** (2002), no. 2 479–502.

[21] S. Kim, R. Pasupathy, and S. G. Henderson, *A guide to sample average approximation*, in *Handbook of simulation optimization*, pp. 207–243. Springer, 2015.

[22] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory.* Society for Industrial and Applied Mathematics, second edition ed., 2014.

[23] G. Calafiore and M. C. Campi, *Uncertain convex programs: randomized solutions and confidence levels*, *Mathematical Programming* **102** (2005), no. 1 25–46.

[24] G. Calafiore and M. C. Campi, *The scenario approach to robust control design*, *IEEE Transactions on automatic control* **51** (2006), no. 5 742–753.

[25] P. Mohajerin Esfahani, T. Sutter, and J. Lygeros, *Performance Bounds for the Scenario Approach and an Extension to a Class of Non-Convex Programs*, *IEEE Transactions on Automatic Control* **60** (Jan., 2015) 46–58. Conference Name: IEEE Transactions on Automatic Control.

[26] A. Ben Tal and A. Nemirovski, *Robust optimization methodology and applications*, *Mathematical Programming* (may, 2002).

[27] V. Gabrel, C. Murat, and A. Thiele, *Recent advances in robust optimization: An overview*, *European Journal of Operational Research* **235** (june, 2014) 471–483.

[28] D. A. Copp and J. P. Hespanha, *Simultaneous nonlinear model predictive control and state estimation*, *Automatica* **77** (2017) 143–154.

[29] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari, *Learning with a Strong Adversary*, .

[30] U. Shaham, Y. Yamada, and S. Negahban, *Understanding adversarial training: Increasing local stability of supervised models through robust optimization*, *Neurocomputing* **307** (sep, 2018) 195–204.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards Deep Learning Models Resistant to Adversarial Attacks*, *arXiv:1706.06083 [cs, stat]* (Sept., 2019). arXiv: 1706.06083.

[32] C. Jin, P. Netrapalli, and M. I. Jordan, *What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?*, arXiv:1902.0061.

[33] K. Margellos, P. Goulart, and J. Lygeros, *On the road between robust optimization and the scenario approach for chance constrained optimization problems*, *IEEE Transactions on Automatic Control* **59** (2014), no. 8 2258–2263.

[34] X. Chen, M. Sim, and P. Sun, *A robust optimization perspective on stochastic programming*, *Operations Research* **55** (2007), no. 6 1058–1071.

[35] C. Bandi and D. Bertsimas, *Tractable stochastic analysis in high dimensions via robust optimization*, *Mathematical programming* **134** (2012), no. 1 23–70.

[36] P. M. Esfahani and D. Kuhn, *Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations*, *Mathematical Programming* **171** (2018), no. 1 115–166.

[37] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh, *Wasserstein distributionally robust optimization: Theory and applications in machine learning*, in *Operations Research & Management Science in the Age of Analytics*, pp. 130–166. INFORMS, 2019.

[38] E. Delage and Y. Ye, *Distributionally robust optimization under moment uncertainty with application to data-driven problems*, *Operations research* **58** (2010), no. 3 595–612.

[39] W. Wiesemann, D. Kuhn, and M. Sim, *Distributionally robust convex optimization*, *Operations Research* **62** (2014), no. 6 1358–1376.

[40] A. Ben-Tal, D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, *Robust Solutions of Optimization Problems Affected by Uncertain Probabilities*, *Management Science* **59** (2013), no. 2 341–357. Publisher: INFORMS.

[41] J. Goh and M. Sim, *Distributionally robust optimization and its tractable approximations*, *Operations research* **58** (2010), no. 4-part-1 902–917.

[42] H. Rahimian and S. Mehrotra, *Distributionally Robust Optimization: A Review*, .

[43] A. Bemporad and M. Morari, *Robust model predictive control: A survey*, in *Robustness in identification and control* (A. Garulli and A. Tesi, eds.), Lecture Notes in Control and Information Sciences, (London), pp. 207–226, Springer, 1999.

[44] D. A. Copp and J. P. Hespanha, *Simultaneous nonlinear model predictive control and state estimation*, *Automatica* **77** (Mar., 2017) 143–154.

[45] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, *Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport*, in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, pp. 125–132, 2008.

[46] G. Yang, R. Poovendran, and J. P. Hespanha, *Adaptive learning in two-player stackelberg games with application to network security*, *arXiv preprint arXiv:2101.03253* (2021).

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Nets*, *Advances in neural information processing systems* **27** (2014) 2672–2680.

[48] R. Chinchilla and J. P. Hespanha, *Optimization-based Estimation of Expected Values with Application to Stochastic Programming*, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, (Nice, France), pp. 6356–6361, IEEE, Dec., 2019.

[49] C. Bandi and D. Bertsimas, *Tractable stochastic analysis in high dimensions via robust optimization*, *Mathematical Programming* **134** (Aug., 2012) 23–70.

[50] B. Colson, P. Marcotte, and G. Savard, *An overview of bilevel optimization*, *Annals of Operations Research* **153** (Sept., 2007) 235–256.

[51] Y. Beck, I. Ljubic, and M. Schmidt, *A Brief Introduction to Robust Bilevel Optimization*, Jan., 2023. arXiv:2211.16072 [math].

[52] C. Jin, P. Netrapalli, and M. I. Jordan, *What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?*, *arXiv:1902.00618 [cs, math, stat]* (Feb., 2019). arXiv: 1902.00618.

[53] Y.-H. Dai and L. Zhang, *Optimality Conditions for Constrained Minimax Optimization*, *CSIAM Transactions on Applied Mathematics* **1** (June, 2020) 296–315. arXiv: 2004.09730.

[54] J. Nocedal and S. J. Wright, *Numerical optimization.* Springer series in operations research. Springer, New York, 2nd ed.. ed., 2006.

[55] J. P. Hespanha, *Tenscalc: a toolbox to generate fast code to solve nonlinear constrained minimizations and compute nash equilibria*, .

[56] J. v. Neumann, *Zur theorie der gesellschaftsspiele*, *Mathematische annalen* **100** (1928), no. 1 295–320.

[57] E. K. Ryu and S. Boyd, *A primer on monotone operator methods*, *APPL. COMPUT. MATH.* (2016).

[58] A. Ben-Tal and A. Nemirovski, *Robust optimization - methodology and applications*, *Mathematical Programming* **92** (May, 2002) 453–480.

[59] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization.* Princeton University Press, Aug., 2009. Google-Books-ID: DttjR7IpjUEC.

[60] D. Bertsimas, D. B. Brown, and C. Caramanis, *Theory and Applications of Robust Optimization*, *SIAM Review* **53** (Jan., 2011) 464–501. Publisher: Society for Industrial and Applied Mathematics.

[61] A. Mutapcic and S. Boyd, *Cutting-set methods for robust convex optimization with pessimizing oracles*, *Optimization Methods and Software* **24** (June, 2009) 381–406.

[62] M. Nouiehed, M. Sanjabi, T. Huang, J. D. Lee, and M. Razaviyayn, *Solving a class of non-convex min-max games using iterative first order methods*, .

[63] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, *Unrolled generative adversarial networks*, arXiv:1611.0216.

[64] A. Mokhtari, A. Ozdaglar, and S. Pattathil, *A Unified Analysis of Extra-gradient and Optimistic Gradient Methods for Saddle Point Problems: Proximal Point Approach*, . ISSN: 2640-3498.

[65] T. Lin, C. Jin, and M. I. Jordan, *Near-optimal algorithms for minimax optimization*, arXiv:2002.0241.

[66] M. Liu, H. Rafique, Q. Lin, and T. Yang, *First-order convergence theory for weakly-convex-weakly-concave min-max problems*, arXiv:1810.1020. version: 3.

[67] A. Nemirovski, *Prox-method with rate of convergence O (1/ t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, .

[68] T. Fiez and L. Ratliff, *Gradient descent-ascent provably converges to strict local minmax equilibria with a finite timescale separation*, arXiv:2009.1482.

[69] 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*, .

[70] J. N. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, *Learning with Opponent-Learning Awareness*, *arXiv:1709.04326 [cs]* (Sept., 2018). arXiv: 1709.04326.

[71] A. Letcher, J. Foerster, D. Balduzzi, T. RocktÃ¤schel, and S. Whiteson, *Stable opponent shaping in differentiable games*, arXiv:1811.0846.

[72] F. Schafer and A. Anandkumar, *Competitive Gradient Descent*, *arXiv:1905.12103 [cs, math]* (June, 2020). arXiv: 1905.12103.

[73] Y. Wang, G. Zhang, and J. Ba, *On solving minimax optimization locally: A follow-the-ridge approach*, arXiv:1910.0751.

[74] T. Fiez, B. Chasnov, and L. J. Ratliff, *Convergence of learning dynamics in stackelberg games*, arXiv:1906.0121.

[75] G. Zhang, K. Wu, P. Poupart, and Y. Yu, *Newton-type methods for minimax optimization*, arXiv:2006.1459.

[76] L. Luo and C. Chen, *Finding second-order stationary point for nonconvex-strongly-concave minimax problem*, arXiv:2110.0481.

[77] Z. Chen, Q. Li, and Y. Zhou, *Escaping saddle points in nonconvex minimax optimization via cubic-regularized gradient descent-ascent*, arXiv:2110.0709.

[78] M. Huang, K. Ji, S. Ma, and L. Lai, *Efficiently escaping saddle points in bilevel optimization*, arXiv:2202.0368.

[79] J. von Neumann, *Zur theorie der gesellschaftsspiele*, *Mathematische annalen* **100** (1928), no. 1 295–320.

[80] K. Fan, *Minimax theorems*, *Proceedings of the National Academy of Sciences of the United States of America* **39** (1953), no. 1 42.

[81] M. Sion *et. al.*, *On general minimax theorems.*, *Pacific Journal of mathematics* **8** (1958), no. 1 171–176.

[82] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, *arXiv:1312.6199 [cs]* (Feb., 2014). arXiv: 1312.6199.

[83] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, *arXiv:1412.6572 [cs, stat]* (Mar., 2015). arXiv: 1412.6572.

[84] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, *DeepFool: a simple and accurate method to fool deep neural networks*, *arXiv:1511.04599 [cs]* (July, 2016). arXiv: 1511.04599.

[85] A. Bemporad and M. Morari, *Robust model predictive control: A survey*, in *Robustness in identification and control* (A. Garulli and A. Tesi, eds.), vol. 245, pp. 207–226. Springer London, London, 1999.

[86] L. Magni and R. Scattolini, *Robustness and robust design of mpc for nonlinear discrete-time systems*, in *Assessment and future directions of nonlinear model predictive control*, pp. 239–254. Springer, 2007.

[87] P. J. Huber, *Robust estimation of a location parameter*, in *Breakthroughs in statistics*, pp. 492–518. Springer, 1992.

[88] R. G. Staudte and S. J. Sheather, *Robust estimation and testing*, vol. 918. John Wiley & Sons, 2011.

[89] R. Chinchilla and J. P. Hespanha, *Optimization-based estimation of expected values with application to stochastic programming*, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 6356–6361, IEEE, 2019.

[90] A. Mutapcic and S. Boyd, *Cutting-set methods for robust convex optimization with pessimizing oracles*, *Optimization Methods & Software* **24** (2009), no. 3 381–406.

[91] D. Bertsimas, I. Dunning, and M. Lubin, *Reformulation versus cutting-planes for robust optimization*, *Computational Management Science* **13** (2016), no. 2 195–217.

[92] J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, *Learning with opponent-learning awareness*, in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 122–130, 2018.

[93] A. Letcher, J. Foerster, D. Balduzzi, T. Rocktäschel, and S. Whiteson, *Stable opponent shaping in differentiable games*, *arXiv preprint arXiv:1811.08469* (2018).

[94] Springer Verlag and European Mathematical Society, *Right-ordered group*, *Encyclopedia of Mathematics.* URL: `URL:http://www.encyclopediaofmath.org/index.php?title=Right-sordered_group&oldid=43497`. Accessed on 2020-04-08.

[95] V. Vapnik, *The nature of statistical learning theory.* Statistics for Engineering and Information Science. Springer, New York, 2nd ed., 1999.

[96] A. Shapiro, *Computational complexity of stochastic programming: Monte carlo sampling approach*, in *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pp. 2979–2995, World Scientific, 2010.

[97] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, *Mathematical Programming* **89** (Nov., 2000) 149–185.

[98] A. Wachter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, *Mathematical Programming* **106** (Mar., 2006) 25–57.

[99] J. P. Hespanha, *TensCalc: A toolbox to generate fast code to solve nonlinear constrained minimizations and compute Nash equilibria.*, tech. rep., Center for Control, Dynamical Systems and Computation; University of California, Santa Barbara, https://github.com/hespanha/tenscalc, 2017.

[100] C. Daskalakis and I. Panageas, *The Limit Points of (Optimistic) Gradient Descent in Min-Max Optimization*, .

[101] K. Ebrahimi, N. Elia, and U. Vaidya, *A continuous time dynamical system approach for solving robust optimization*, in *2019 18th European Control Conference (ECC)*, pp. 1479–1485, 06, 2019.

[102] M. Razaviyayn, T. Huang, S. Lu, M. Nouiehed, M. Sanjabi, and M. Hong, *Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances*, . Conference Name: IEEE Signal Processing Magazine.

[103] H. Rafique, M. Liu, Q. Lin, and T. Yang, *Non-convex min-max optimization: Provable algorithms and applications in machine learning*, arXiv:1810.0206.

[104] S. Lu, I. Tsaknakis, M. Hong, and Y. Chen, *Hybrid block successive approximation for one-sided non-convex min-max problems: Algorithms and applications*, . Conference Name: IEEE Transactions on Signal Processing.

[105] E. V. Mazumdar, M. I. Jordan, and S. S. Sastry, *On finding local nash equilibria (and only local nash equilibria) in zero-sum games*, arXiv:1901.0083.

[106] E. A. Ok, *Real analysis with economic applications / Efe A. Ok.* University Press, 2007.

[107] D. P. Bertsekas, *Nonlinear Programming.* Athena,1999. 2nd Edition, 2nd edition ed., 1999.

[108] S. Quintero, D. A. Copp, and J. P. Hespanha, *Robust coordination of small UAVs for vision-based target tracking using output-feedback MPC with MHE*, in *Cooperative Control of Multi-agent Systems: Theory and Applications* (E. Garcia, Y. Wang, D. Casbeer, and F. Zhang, eds.). Wiley & Sons, Hoboken, NJ, 2017.

[109] D. A. Copp, *Simultaneous Nonlinear Model Predictive Control and State Estimation: Theory and Applications.* PhD thesis, University of California, Santa Barbara, CA, USA, Nov., 2016.

[110] G. J. McLachlan, *The EM algorithm and extensions.* Wiley series in probability and statistics. Wiley-Interscience, 2nd ed.. ed., 2008.

[111] C. Abraham, G. Biau, and B. Cadre, *Simple Estimation of the Mode of a Multivariate Density*, *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* **31** (2003), no. 1 23–34. Publisher: [Statistical Society of Canada, Wiley].

[112] K. Andersson, I. Kaminer, V. Dobrokhodov, and V. Cichella, *Thermal Centering Control for Autonomous Soaring; Stability Analysis and Flight Test Results*, *Journal of Guidance, Control, and Dynamics* **35** (may, 2012) 963–975.

[113] D. Edwards, *Implementation Details and Flight Test Results of an Autonomous Soaring Controller*, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, aug, 2008.

[114] V. N. Dobrokhodov, N. Camacho, and K. D. Jones, *Cooperative Autonomy of Multiple Solar-Powered Thermaling Gliders*, *IFAC Proceedings Volumes* **47** (jan, 2014) 1222–1227.

[115] J. P. Hespanha, *Linear systems theory.* University Press, Princeton, 2 ed., 2018.

[116] F. Zhang, ed., *The Schur complement and its applications.* No. v. 4 in Numerical methods and algorithms. Springer, New York, 2005.

[117] R. J. Vanderbei, *Symmetric quasidefinite matrices*, . Publisher: Society for Industrial and Applied Mathematics.

[118] N. J. Higham and S. H. Cheng, *Modifying the inertia of matrices arising in optimization*, .

[119] L. Adolphs, H. Daneshmand, A. Lucchi, and T. Hofmann, *Local Saddle Point Optimization: A Curvature Exploitation Approach*, *arXiv:1805.05751 [cs, math, stat]* (Feb., 2019). arXiv: 1805.05751.

[120] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model predictive control: theory, computation, and design.* Nob Hill Publishing, Madison, Wisconsin, 2nd edition ed., 2017. OCLC: 1020170256 Citation Key Alias: rawlingsModelPredictiveControl2017a.

[121] A. Shapiro, *Differentiability Properties of Metric Projections onto Convex Sets*, *Journal of Optimization Theory and Applications* **169** (June, 2016) 953–964.

[122] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 3rd ed., 2016.