PSL UNIVERSITÉ PARIS

HABILITATION À DIRIGER DES RECHERCHES

DE L'UNIVERSITÉ PSL

Présentée à l'Université Paris Dauphine-PSL

Algorithmes pour l'optimisation non convexe et leurs bornes de complexité

/

Nonconvex optimization algorithms with complexity guarantees

Présentation des travaux par

Clément ROYER

Le 22 septembre 2025

Discipline

Informatique

Composition du jury :

Anne AUGER Directrice de recherches INRIA	*Présidente*
Alexandre D'ASPREMONT Directeur de recherches CNRS & DI ENS	*Rapporteur*
Coralia CARTIS Professeure University of Oxford	*Rapporteure*
Edouard PAUWELS Professeur Toulouse School of Economics	*Rapporteur*
Antonin CHAMBOLLE Directeur de recherches CNRS & Université Paris Dauphine-PSL	*Coordinateur*

Dauphine | PSL UNIVERSITÉ PARIS

Contents

Ackowledgments

I would first like to thank Antonin Chambolle for agreeing to act as coordinator for this habilitation thesis, and I look forward to the next steps of our collaboration. I am also thankful to Coralia Cartis, Alexandre d'Aspremont and Edouard Pauwels for agreeing to act as referees for this manuscript. Their work has been an inspiration to me, and their feedback means a lot. Last but not least, I thank Anne Auger for acting as chair of the committee, and for several kind exchanges throughout the years.

Postdoc years seem like a long time ago, but I will never forget how amazing those years were, and how important working in Wisconsin has been to my career and my life. Thanks to Stephen Wright for this opportunity, as well as Michael O'Neill for all the good times we had then. On a more personal note, thanks to my friends from the Forward! Marching Band for bringing music back into the picture.

From the US to Paris, I have been extremely lucky to collaborate with a number of people having various backgrounds and seniority levels, and I am very grateful for what it brought me on both a research and a humane level. My co-author list is starting to lengthen, so I won't list all of you, and simply thank you again.

All my colleagues in Dauphine, faculty and staff, have taught me what it is to be an academic. Special thanks to my office mate Alexandre Allauzen, to Irène Waldspurger for all of our discussions, and to Florian Yger for sharing so much throughout the years. Everyone at LAMSADE (beyond MILES and Pôle 2) and many at Dauphine and PSL (way beyond departments) would deserve to be mentioned here, be sure that I had a thought for you while writing these lines. In particular, I remember quite fondly the time I shared with my postdocs Annette Dumas and Florentin Goyens, and I look forward to our next discussions.

As I am writing these lines, I think back to the students that I worked with during my time in Dauphine, that I taught to, and I realize how much I learned from them. Iskander, Sébastien and Bastien have taught me what it means to be a supervisor, and I am sure that I will learn from Gaetano as well. Rémi Chan–Renous-Legoubin was my student in a C++ class, and we ended up writing a paper together as a result of his internship. Even though other internships did not lead to a publication, every one of them (Thomas Georges, Marc Kaspar, Christian Kayo, Eloi Martin, Luca Solbiati) was an opportunity to initiate students to research environments, and I appreciate them all.

Last but not least, I thank Claire, Jules and Étienne, with all my love, for teaching me every day what it means to be a partner and a dad.

Habilitation soundtrack

Bill Laurance - *Live At Ronnie Scott's*, *Live at the Philharmonie Cologne (with WDR Big Band)*, *Affinity*, *Do What You Want*.

Bill Laurance and The Untold Orchestra - *Live at EFG London Jazz Festival*, *Cables Rewired*, *Bloom*.

L'impératrice - *L'Odyssée*, *Matahari*, *Tako Tsubo*, *Pulsar*, *Live @ La Felicità for Drop 2019*, *Live Grand Palais 2021*, *Live Palais Bulles 2024*, *Live @ Flow Festival Helsinki 2024*.

Incognito - *Live in London: The 35th Anniversary Show*, *Tomorrow's New Dream*, *Always There 1981-2021: 40 Years and Still Groovin'*, *Into You*.

Léon Phal Quintet - *Canto Bello*, *Live Tremplin Grand Est 2019*, *Live TSF Jazz 2020*, *Dust to stars*, *Stress killer*, *Stress Killer Deluxe Edition*, *No Pain No Champagne*.

Simon Stålenhag - *The Electric State*.

Snarky Puppy - *The Only Constant*, *The World Is Getting Smaller*, *Bring Us The Bright*, *Empire Central*, *Live in Paris 2023*.

Cory Wong - *Motivational Music for the Syncopated Soul*, *The Lucky One*, *Live in Jazz in Marciac 2023*, *Live at Montreux Jazz Festival 2023*, *The Power Station Tour (West Coast)*, *Wong Air (Live in America)*, *Live at L'Olympia*.

Cory Wong with Metropol Orkest - *Starship Syncopation*.

Chapter 1

Introduction

Optimization is a research field concerned with making the best decision out of a set of alternatives. Numerical optimization is concerned with building algorithms for solving optimization problems. These algorithms should not only be mathematically certified to reach the desired (approximate) solution, but they should also be implementable on a computer, and demonstrate their efficiency in practice. For this reason, researchers have sought algorithms that can converge to a problem solution (or, more often, an approximation thereof) at the lowest possible cost. This paradigm is that of *complexity analysis*, and is now of primary importance in optimization theory [149]. Although complexity results were mostly used in convex optimization until the early 2000s [116, 118], the prevalence of nonconvex optimization in modern applications such as machine learning generated significant developments starting in the 2010s [33].

A common theme in most of the author's research is the derivation of complexity results for nonconvex optimization algorithms. The manuscript summarizes this research through this viewpoint. After a review of the associated literature in Section 1.1, the contents of the manuscript are described in Section 1.2.

1.1 Complexity results in nonconvex optimization

1.1.1 Complexity analysis in optimization

Early results in complexity date back to linear programming, and were instrumental to the interior-point revolution that occurred at the end of the twentieth century [147]. For nonlinear optimization, oracle complexity models became prevalent in the 1970s, with the terminology being set with the textbook of Nemirovski and Yudin [116]. Prompted by the breakthrough results of Polyak[Heavy-ball] and Nesterov [117], numerous first-order methods were proposed and analyzed from a complexity perspective. Understanding the acceleration phenomenon, that gave rise to optimal complexity algorithms, proved key to obtaining the best known results. The field of convex optimization has now reached a mature stage [120, 55], although it remains a popular topic of investigation.

Evaluation complexity results for nonconvex optimization were significantly less investigated until the late 2000s, despite early results for gradient descent [118]. Following the seminal work of Nesterov and Polyak [121], a number of algorithmic variants were analyzed from an iteration and evaluation complexity perspective. Classical variants of Newton's method proved to be as slow as gradient descent in the worst case [28], while cubic regularization methods exhibited optimal complexity among second-order algorithms [121]. A number of variants on this paradigm were then

analyzed [29, 30, 31, 32], and the field of complexity in nonconvex optimization is now established, as exemplified by the textbook of Cartis, Gould and Toint [33].

1.1.2 Main complexity questions in smooth nonconvex optimization

In this manuscript, we mainly consider smooth nonconvex optimization problems of the form

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\boldsymbol{x}), \tag{1.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth (i.e. at least continuously differentiable) nonconvex function. Owing to the nonconvexity of f, finding global or even local minima of f is intractable in general [114]. For this reason, a more natural goal consists in searching for points that satisfy approximate stationary conditions.

Given a tolerance $\epsilon \in (0, 1)$[1], an *ϵ-first order stationary point* of problem (1.1.1) is a vector $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ such that

$$\|\nabla f(\boldsymbol{x})\| \leq \epsilon, \tag{1.1.2}$$

where $\|\cdot\|$ denotes the Euclidean (or ℓ_2) norm on $\mathbb{R}^n$.

Similarly, assuming that the function f is $\mathcal{C}^2$ and given two tolerances $\epsilon, \epsilon_H \in (0, 1)^2$, an *$(\epsilon, \epsilon_H)$-second order stationary point* of problem (1.1.1) is a vector $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ such that

$$\|\nabla f(\boldsymbol{x})\| \leq \epsilon \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(\boldsymbol{x})) \succeq -\epsilon_H \mathbf{I}_n, \tag{1.1.3}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the matrix and $\boldsymbol{I}_n$ is the identity matrix in $\mathbb{R}^{n\times n}$.

In this manuscript, complexity analysis is concerned with one of the two following questions.

Question 1.1.1 *Given an algorithm applied to problem (1.1.1) and $\epsilon \in (0, 1)$, what is the worst-case cost (in terms of iterations, evaluations, etc) of that algorithm to reach an ϵ-first order stationary point?*

Question 1.1.2 *Given an algorithm applied to problem (1.1.1) and $\epsilon \in (0, 1)$, what is the worst-case cost (in terms of iterations, evaluations, etc) of that algorithm to reach an (ϵ, ϵ_H)-second order stationary point?*

Answering either Question 1.1.1 or Question 1.1.2 requires a careful study of algorithmic behavior, and appropriate metrics of cost.

1.2 Manuscript roadmap

This manuscript aims at providing an overview of the author's research through the lens of complexity guarantees in nonconvex optimization. In selecting the material to be presented, the author aimed at highlighting his main contributions conducted both as an independent researcher and as a faculty advisor.

[1] For simplicity, we will always assume that tolerances are smaller than 1. In addition to simplifying the analysis by putting the emphasis on small values of ϵ, this also reflects popular practice that scales the objective function by a constant to guarantee that initial gradients are of norm 1.

1.2.1 Contributions

The rest of this manuscript is organized along four chapters. Chapter 2 to 4 follow a similar template. The first section of these chapters serves an introductory purpose. It reviews complexity results relevant for the chapter's setting, by putting an emphasis on a particular technique which the other sections build upon. The second section describes a major contribution from the author conducted either during his postdoctoral studies or as a faculty researcher. The third section highlights a more recent contribution involving a junior researcher (master student, PhD student, or postdoctoral researcher) that worked under the author's supervision. Those three chapters also include a final section that highlights additional work and short-term research perspectives for the chapter's line of work. For simplicity, a specific family of algorithms is used to describe the key concepts and contributions, while other algorithmic frameworks are discussed throughout.

Chapter 5 concludes the manuscript by reflecting on the research conducted by the author, and draws on its current environment to provide a long-term vision for the future research conducted by the author or his research group.

1.2.2 Notations and terminology

- Scalars (i.e. reals) are denoted by lowercase letters: $a, b, c, \alpha, \beta, \gamma$.
- Vectors are denoted by **bold** lowercase letters: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$.
- Matrices are denoted by **bold** uppercase letters: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$.
- Sets are denoted by **bold** uppercase cursive letters : $\mathcal{A}, \mathcal{B}, \mathcal{C}$.
- The set of natural numbers (nonnegative integers) is denoted by $\mathbb{N}$; the set of integers is denoted by $\mathbb{Z}$.
- The set of real numbers is denoted by $\mathbb{R}$. Our notations for the subset of nonnegative real numbers and the set of positive real numbers are $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively.
- The notation $\mathbb{R}^n$ is used for the set of vectors with $n \in \mathbb{N}$ real components; although we may not explicitly state it, we always assume that $n \geq 1$.
- A vector $\boldsymbol{x} \in \mathbb{R}^n$ is thought as a column vector, with $x_i \in \mathbb{R}$ denoting its i-th coordinate in the canonical basis of $\mathbb{R}^n$. We thus write $\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, or, in a compact form, $\boldsymbol{x} = [x_i]_{1 \leq 1 \leq n}$.
- Given a column vector $\boldsymbol{x} \in \mathbb{R}^n$, the corresponding row vector is denoted by $\boldsymbol{x}^{\mathrm{T}}$, so that $\boldsymbol{x}^{\mathrm{T}} = [x_1 \ \cdots \ x_n]$ and $[\boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}} = \boldsymbol{x}$. The scalar product between two vectors in $\mathbb{R}^n$ is defined as $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = \sum_{i=1}^n x_i y_i$.
- The Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is defined by $\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$.
- For any integer $n \geq 1$, the vectors $\mathbf{0}_n$ and $\mathbf{1}_n$ correspond to the vectors of $\mathbb{R}^n$ for which all elements are 0 or 1, respectively.

- We use $\mathbb{R}^{n\times d}$ to denote the set of real rectangular matrices with n rows and d columns, where n and d will always be assumed to be at least 1. If $n = d$, $\mathbb{R}^{n\times n}$ refers to the set of square matrices of size n.
- We identify a matrix in $\mathbb{R}^{n\times 1}$ with its corresponding column vector in $\mathbb{R}^n$.
- Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$, $\boldsymbol{A}_{ij}$ refers to the coefficient from the i-th row and the j-th column of $\boldsymbol{A}$: the diagonal of $\boldsymbol{A}$ is given by the coefficients $\boldsymbol{A}_{ii}$. Provided this notation is not ambiguous, we use the notations $\boldsymbol{A}$, $[\boldsymbol{A}_{ij}]_{\substack{1\leq i\leq n\\ 1\leq j\leq d}}$ and $[\boldsymbol{A}_{ij}]$ interchangeably.
- Depending on the context, we may use $\boldsymbol{a}_i^{\mathrm{T}}$ to denote the i-th row of $\boldsymbol{A}$ or $\boldsymbol{a}_j$ to denote the j-th column of $\boldsymbol{A}$, leading to $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{a}_n^{\mathrm{T}} \end{bmatrix}$ or $\boldsymbol{A} = [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_d]$, respectively.
- Given $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathbb{R}^{n\times d}$, the Frobenius norm of $\boldsymbol{A}$ will be denoted by $\|\boldsymbol{A}\|_F^2 := \sum_{i=1}^n \sum_{j=1}^d \boldsymbol{A}_{ij}$.
- For every $n \geq 1$, $\boldsymbol{I}_n$ refers to the identity matrix in $\mathbb{R}^{n\times n}$ (with 1s on the diagonal and 0s elsewhere). The notations $\boldsymbol{0}_{n,d}$ and $\boldsymbol{1}_{n,d}$ will be used for matrices in $\mathbb{R}^{n\times d}$ that consist solely of 0s or 1s, respectively.
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is called $\mathcal{C}^1$ if it is continuously differentiable, and $\mathcal{C}^{1,1}$ if it is continuously differentiable with a Lipschitz continuous gradient. The notation $\mathcal{C}^{1,1}_{L_{\mathrm{g}}}$ will denote the set of $\mathcal{C}^{1,1}$ functions with L_{g}-Lipschitz continuous gradient.
- A function $f : \mathbb{R}^n \to \mathbb{R}$ is called $\mathcal{C}^2$ if it is twice continuously differentiable, and $\mathcal{C}^{2,2}$ if it is continuously differentiable with a Lipschitz continuous Hessian.The notation $\mathcal{C}^{2,2}_L$ will denote the set of $\mathcal{C}^{2,2}$ functions with L-Lipschitz continuous Hessian.
- Throughout the manuscript, $\mathcal{O}(E)$ will denote $C\,E$, where C is a positive constant that does not depend on the quantities of interest arising in E. Similarly, $\tilde{\mathcal{O}}(E)$ indicates that the constant depends at most logarithmically on the quantities of interest in E.

Chapter 2

Complexity of direct-search methods

This chapter is concerned with derivative-free optimization (DFO), wherein a function must be optimized while relying solely on function evaluations. Mathematically, we consider the problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\boldsymbol{x}), \tag{2.0.1}$$

where only accesses to an oracle f are available, regardless of the smoothness properties of the objective. A typical occurrence of such problems consists in optimizing the parameters of a simulation code, that is both expensive to run and cannot be differentiated automatically, either due to the use of proprietary software or that of poorly written subroutines. The field of derivative-free optimization, also called blackbox optimization, was developed on this paradigm starting as early as the 1960s, and grew out to become a subfield of optimization in itself [7, 45, 103]. The main optimization conferences now always feature a derivative-free optimization stream, while zeroth-order algorithms (a special class of derivative-free algorithms that approximate gradients through finite differences formulas) have also gained popularity in the data science and learning communities.

This chapter begins with a brief overview of complexity results in derivative-free optimization, with a particular focus on results derived for direct-search techniques (Section 2.1). Using again the setup of direct search, the area of derivative-free methods based on random subspaces is introduced, along with the possible complexity benefits that were demonstrated by the author together with Lindon Roberts (Section 2.2). Section 2.3.1, based on contributions by the author and PhD student Sébastien Kerleau, takes a closer look at quantities involved in the complexity bounds for deterministic direct search, with a focus on how those quantities can be computed.

2.1 Direct-search methods and complexity results

In this section, we review complexity results in derivative-free optimization in the nonconvex setting through the lens of direct-search methods, the class of algorithms that was most studied by the author. Section 2.1.1 presents a basic direct-search algorithm, that is used to derive complexity guarantees in Section 2.1.2. Complexity guarantees for other derivative-free techniques are discussed further in Section 2.1.3.

2.1.1 Direct-search algorithm and main concepts

Direct-search methods are arguably the simplest class of derivative-free algorithms, in that they rely on direct exploration of the variable space through suitably chosen directions. Under relatively mild

assumptions on those directions, convergence guarantees can be derived for a number of variants on the direct-search paradigm [7, 99]. However, only a subset of those variants can be equipped with complexity guarantees [103, 61].

Algorithm 1 describes a basic direct-search method of so-called directional type, that can be endowed with complexity guarantees. At each iteration, a set of directions $\mathcal{D}_j$ is considered for possible evaluation of f in that direction. If one of those directions satisfies the *sufficient decrease condition* (2.1.1), the corresponding point is accepted and the stepsize parameter α_j is (possibly) increased. Otherwise, the current iterate is kept for the next iteration and the stepsize is decreased.

Algorithm 1: A basic direct-search algorithm.

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_{\max} > 0$, $\alpha_0 \in (0, \alpha_{\max}]$, $c > 0$, $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$, $m \in \mathbb{N}$.
for $j = 0, 1, \ldots$ **do**
  Compute a set $\mathcal{D}_j \subset \mathbb{R}^n$ of m vectors.
  If there exists $\boldsymbol{d}_j \in \mathcal{D}_j$ such that

$$f(\boldsymbol{x}_j + \alpha_j \boldsymbol{d}_j) < f(\boldsymbol{x}_j) - \frac{c}{2}\alpha_j^2 \|\boldsymbol{d}_j\|^2, \tag{2.1.1}$$

  set $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j + \alpha_j \boldsymbol{d}_j$ and $\alpha_{j+1} := \min\{\gamma_{\text{inc}}\alpha_j, \alpha_{\max}\}$.
  Otherwise, set $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j$ and $\alpha_{j+1} := \gamma_{\text{dec}}\alpha_j$.
end

An important feature of Algorithm 1 is that an iteration can end *as soon as* a direction of sufficient decrease has been found. This process, termed opportunistic polling in the literature, is often key to practical efficiency. From a complexity perspective, however, each iteration has a cost of $|\mathcal{D}_j|$ function evaluations in the worst case.

In a smooth setting, Algorithm 1 requires the *polling directions* $\mathcal{D}_j$ to form a positive spanning set, in the following sense [56].

Definition 2.1.1 *A set $\mathcal{D}$ of vectors in $\mathbb{R}^n$ is called a* positive spanning set *if it spans $\mathbb{R}^n$ by nonnegative linear combinations.*

Positive spanning sets have been used to endow direct-search methods with global convergence guarantees since the late 1990s [108] and are instrumental to modern analysis of those methods [7, 99]. This analysis relies on a specific quantity called the *cosine measure*, and defined by

$$\text{cm}(\mathcal{D}) := \min_{\substack{\boldsymbol{v}\in\mathbb{R}^n \\ \boldsymbol{v}\neq\boldsymbol{0}}} \max_{\substack{\boldsymbol{d}\in\mathcal{D} \\ \boldsymbol{d}\neq\boldsymbol{0}}} \frac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{d}}{\|\boldsymbol{v}\|\,\|\boldsymbol{d}\|}. \tag{2.1.2}$$

The cosine measure of an arbitrary set measures how well this set approximates any vector in the space. Moreover, a set $\mathcal{D}$ is a PSS of $\mathbb{R}^n$ if and only if $\text{cm}(\mathcal{D}) > 0$ [7]. Choosing good polling sets for optimization then amounts to a compromise between the size of the PSS and the cosine measure. Both aspects have been explored by the author in the context of his work, and will be detailed in Sections 2.2 and 2.3.

2.1.2 Complexity analysis of direct search

Vicente [144] was the first to derive complexity results for direct-search methods. The sufficient decrease condition (2.1.1) is instrumental to obtaining complexity results. We present such a result below in the smooth setting.

Assumption 2.1.1 *The function f is continuously differentiable with L-Lipschitz continuous gradient, where $L > 0$.*

Assumption 2.1.2 *The function f is bounded below by $f_{\text{low}} \in \mathbb{R}$.*

We also make the following assumption on the polling sets used by the algorithm.

Assumption 2.1.3 *At every iteration j of Algorithm 1, the polling set $\mathcal{D}_k$ satisfies the following properties:*

(i) $\operatorname{cm}(\mathcal{D}_j) \geq \tau$ *for some* $\tau \in (0, 1)$.

(ii) $\forall \boldsymbol{d} \in \mathcal{D}_j,\ d_{\min} \leq \|\boldsymbol{d}\| \leq d_{\max}$, *where* $0 < d_{\min} \leq d_{\max}$.

The first property implies that all sets are positive spanning sets, and that the sequence of cosine measures cannot vanish in the limit. Choosing $\mathcal{D}_j = \mathcal{D}$ for all j with $\mathcal{D}$ a positive spanning set clearly satisfies this property, however efficient direct-search techniques change PSSs at every iteration [99]. The second property can easily be guaranteed by normalizing directions to ensure their norm is bounded. A common choice for that purpose consists in using unit vectors.

A complexity proof for Algorithm 1 can be obtained as follows. On one hand, Assumptions 2.1.1 and 2.1.3 guarantee that

$$\alpha_j \leq \mathcal{O}(\|\nabla f(\boldsymbol{x}_j)\|) \tag{2.1.3}$$

for every unsuccessful iteration, at which no direction of sufficient decrease has been found. On the other hand, Assumption 2.1.2 together with the sufficient decrease condition (2.1.1) as well as the updating rules for α_j guarantee that

$$\sum_{j=0}^{\infty} \alpha_j^2 < \infty, \tag{2.1.4}$$

implying in particular that $\alpha_j \to 0$. Combining those results eventually lead to the following complexity bound [100, 144].

Theorem 2.1.1 *Let Assumptions 2.1.1 and 2.1.2 hold. Suppose that Algorithm 1 is applied with polling sets satisfying Assumption 2.1.3. Then, for any $\epsilon \in (0, 1)$, the algorithm computes an iterate x_J such that $\|\nabla f(x_J)\| \leq \epsilon$ in at most*

$$\mathcal{O}\left(\tau^{-2}\,\epsilon^{-2}\right) \tag{2.1.5}$$

iterations and

$$\mathcal{O}\left(m\,\tau^{-2}\,\epsilon^{-2}\right) \tag{2.1.6}$$

function evaluations.

The constants in $\mathcal{O}(\cdot)$ depend on α_0, $f(\boldsymbol{x}_0) - f_{\text{low}}$, γ_{dec}, γ_{inc}, c and L.

Two dependencies are worth highlighting in the bounds (2.1.5) and (2.1.6). First, both bounds depend on τ. The larger τ is, the better the PSSs $\mathcal{D}_k$ are at approximating the entire space. Second, the evaluation complexity bound (2.1.6) depends on m, i.e. the size of the PSSs used at every iteration.

2.1.3 Other complexity results in derivative-free optimization

Model-based derivative algorithms have also been endowed with complexity results on nonconvex problems. In particular, model-based trust-region methods enjoy an iteration complexity in $\mathcal{O}(\kappa^2\,\epsilon^{-2})$, where κ measures the quality of the models that are used. Typical values for κ are of order $\sqrt{n}$, yielding an iteration complexity bound that matches direct search in terms of dependencies on n and ϵ. The evaluation complexity bound can also be shown to be $\mathcal{O}(n^2\epsilon^{-2})$ [64].

Methods rely on finite-difference estimates of derivatives have also been analyzed from a complexity perspective. Cubic regularization based on finite-difference estimates have been endowed with iteration and evaluation complexity guarantees of $\mathcal{O}(n\epsilon^{-3/2})$ and $\mathcal{O}(n^2\log(\epsilon^{-1})\epsilon^{-3/2})$, respectively [30]. This rate improves over direct search in terms of the dependency on ϵ, but not in terms of dimension dependency. A random gradient-free method due to Nesterov [119][1] was shown to satisfy an $\mathcal{O}(n\epsilon^{-2})$ complexity bound in terms of both iterations and evaluations, by relying on randomized finite differences. More recently, several algorithms with improved complexity in terms of dependency on the dimension have been proposed. Quadratic regularization methods with finite-difference estimates have been endowed with an evaluation complexity in $\mathcal{O}(n\epsilon^{-2})$, thereby improving over the complexity of classical direct search [71, 72]. A bound in $\mathcal{O}(n\,\epsilon^{-2})$ can also be obtained for derivative-free line-search methods, that perform extrapolation line search along polling directions [21].

2.2 Dimensionality reduction and complexity

Derivative-free methods have traditionally been used for relatively small-dimensional problems, with dimensions ranging from 100 (for direct-search schemes) to 1000 (for model-based schemes). A partial explanation lies in the necessary use of a number of evaluations that scales with the dimension. For this reason, and motivated by successes in compressed sensing, randomized approaches were introduced to reduce the number of function evaluations used per iteration without compromising convergence guarantees [9, 10, 77].

This section describes the work conducted by the author in that area as a faculty researcher, together with Lindon Roberts. Section 2.2.1 adapts the direct-search framework of Section 2.1.1 to explore low-dimensional subspaces during the iteration process. Complexity guarantees for this method are given in Section 2.2.2. A short literature review on this quickly expanding field is provided in Section 2.2.3.

2.2.1 A subspace perspective on direct search

As in Section 2.1, we are concerned with solving problem (2.0.1) using derivative-free optimization algorithms, and direct-search methods in particular. In a departure from earlier results, however, we will consider randomized versions of the algorithm based on exploring directions within a randomly generated subspace at every iteration.

Algorithm 2 is an adaptation of Algorithm 1 that emphasizes the subspace aspects of this method. At every iteration, one first draws a random matrix $\boldsymbol{P}_j$ that corresponds to a projection onto a low-dimensional subspace (of dimension at most $r \leq n$). A polling set $\mathcal{D}_j$ is then defined within that subspace, implying that the directions considered for polling are $\{\boldsymbol{P}_j^{\mathrm{T}}\boldsymbol{d} \,|\, \boldsymbol{d} \in \mathcal{D}_j\}$. The rest of the algorithm proceeds as in Algorithm 1.

[1] A revised version of this paper was published in 2017 [122].

Algorithm 2: Direct search in random subspaces.

Inputs: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_{\max} > 0$, $\alpha_0 \in (0, \alpha_{\max}]$, $c > 0$, $0 < \gamma_{\text{dec}} < 1 < \gamma_{\text{inc}}$, $m \in \mathbb{N}$; $r \leq n$.
for $j = 0, 1, \ldots$ **do**
Compute a matrix $\boldsymbol{P}_j \in \mathbb{R}^{r \times n}$.
Compute a set $\mathcal{D}_j \subset \mathbb{R}^r$ of m vectors.
If there exists $\boldsymbol{d}_j \in \mathcal{D}_j$ such that

$$f(\boldsymbol{x}_j + \alpha_k \boldsymbol{P}_j^{\mathrm{T}} \boldsymbol{d}_j) < f(\boldsymbol{x}_j) - \frac{c}{2}\alpha_j^2 \|\boldsymbol{P}_j^{\mathrm{T}} \boldsymbol{d}_j\|^2, \tag{2.2.1}$$

set $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j + \alpha_j \boldsymbol{d}_j$ and $\alpha_{j+1} := \min\{\gamma_{\text{inc}} \alpha_j, \alpha_{\max}\}$.
Otherwise, set $\boldsymbol{x}_{j+1} := \boldsymbol{x}_j$ and $\alpha_{j+1} := \gamma_{\text{dec}} \alpha_j$.
end

By setting $r = n$ and $\boldsymbol{P}_j = \boldsymbol{I}_n$ for every j, one recovers the classical direct-search framework. The framework of Algorithm 2 is broader, in that it allows for selecting directions within a proper subspace of $\mathbb{R}^n$. A joint work with Lindon Roberts [136] provided numerical evidence that using $r = 1$ and $\mathcal{D}_j = \{1, -1\}$ and $\boldsymbol{P}_j$ is a vector following a Gaussian distribution is a suitable choice. Note that the theory for Gaussian vectors was not covered from a theoretical perspective by previous works [77].

2.2.2 Probabilistic complexity analysis

We now aim at extending the analysis of Section 2.1.2 to Algorithm 2. To this end, we enforce properties on both the subspace matrices $\{\boldsymbol{P}_j\}$ and the direction sets $\{\mathcal{D}_j\}$. Since we aim at using randomly generated matrices, those properties will only hold with high probability.

We first consider the subspace matrices, where dimensionality reduction arguments can be leveraged. Indeed, our hope in using those matrices is that they define subspaces in which most of the gradient norm (our metric of interest) is preserved. In fact, we only require this property to hold with sufficiently high probability.

Definition 2.2.1 *Let η, σ and $P_{\max}$ be positive quantities. For any realization of Algorithm 2 and any $j \in \mathbb{N}$, the matrix $\boldsymbol{P}_j$ is called* $(\eta, \sigma, P_{\max})$-well aligned *for f at $\boldsymbol{x}_j$ provided*

$$\|\boldsymbol{P}_j \nabla f(\boldsymbol{x}_j)\| \geq \eta \|\nabla f(\boldsymbol{x}_j)\|, \tag{2.2.2}$$

$$\|\boldsymbol{P}_j\| \leq P_{\max}, \tag{2.2.3}$$

$$\sigma_{\min}(\boldsymbol{P}_j) \geq \sigma, \tag{2.2.4}$$

where $\sigma_{\min}(\cdot)$ denotes the minimum nonzero singular value of the matrix $\boldsymbol{P}_j$.

Definition 2.2.2 *The sequence $\{\boldsymbol{P}_j\}_j$ generated by Algorithm 2 is called* $(\eta, \sigma, P_{\max}, q)$-well-aligned *for $q \in (0, 1]$ if*

$$\begin{aligned} & \mathbb{P}\left(\boldsymbol{P}_0 \text{ is } (\eta, \sigma, P_{\max})\text{-well aligned}\right) \geq q \\ \forall k \geq 1, \qquad & \mathbb{P}\left(\boldsymbol{P}_j \text{ is } (\eta, \sigma, P_{\max})\text{-well aligned} \mid \mathcal{F}_{j-1}\right) \geq q, \end{aligned} \tag{2.2.5}$$

where $\mathcal{F}_{j-1}$ is the σ-algebra generated by $\boldsymbol{P}_0, \mathcal{D}_0, \ldots, \boldsymbol{P}_{j-1}, \mathcal{D}_{j-1}$.

Our requirement on $\mathcal{D}_j$ is given below, and is similar to that used in Section 2.1.2. A key difference lies in the use of a cosine measure with respect to a single vector rather than the entire space. Using random directions and lower-dimensional subspaces does not guarantee the positive spanning property. However, obtaining a descent direction is sufficient to guarantee convergence, and it even suffices for $\mathcal{D}_j$ to contain a descent direction in probability [77].

Definition 2.2.3 *Let $\tau \in (0,1]$ and $d_{\max} > 1$. For any realization of Algorithm 2 and any index $j \in \mathbb{N}$, the set $\mathcal{D}_j$ is called $(\tau, d_{\max})$-descent for f and $\boldsymbol{P}_j$ at $\boldsymbol{x}_j$ if*

$$\operatorname{cm}(\mathcal{D}_j, -\boldsymbol{P}_j \nabla f(x_j)) = \max_{\boldsymbol{d} \in \mathcal{D}_j} \frac{-\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_j \nabla f(\boldsymbol{x}_j)}{\|\boldsymbol{d}\| \|\boldsymbol{P}_j \nabla f(\boldsymbol{x}_j)\|} \geq \kappa \tag{2.2.6}$$

and

$$\forall \boldsymbol{d} \in \mathcal{D}_j, \quad d_{\max}^{-1} \leq \|\boldsymbol{d}\| \leq d_{\max}. \tag{2.2.7}$$

Examples of sets satisfying Definition 2.2.3 are positive spanning sets in $\mathbb{R}^r$ with unitary elements. As for the properties of $\boldsymbol{P}_j$, we provide a probabilistic counterpart of Definition 2.2.3 below.

Definition 2.2.4 *The sequence $\{\mathcal{D}_j\}_j$ generated by Algorithm 2 is called* $(\tau, d_{\max}, p)$-descent *for $p \in (0,1]$ if*

$$\begin{array}{rll} & \mathbb{P}\left(\mathcal{D}_0 \text{ is } (\tau, d_{\max})\text{-descent} \middle| \mathcal{F}_{-1/2}\right) & \geq p \\ \forall j \geq 1, & \mathbb{P}\left(\mathcal{D}_j \text{ is } (\tau, d_{\max})\text{-descent} \middle| \mathcal{F}_{j-1/2}\right) & \geq p, \end{array} \tag{2.2.8}$$

where $\mathcal{F}_{j-1/2}$ is the σ-algebra generated by $\boldsymbol{P}_0, \mathcal{D}_0, \ldots, \boldsymbol{P}_{j-1}, \mathcal{D}_{j-1}, \boldsymbol{P}_j$ and $\mathcal{F}_{-1/2}$ is the σ-algebra generated by $\boldsymbol{P}_0$.

Using submartingale arguments, we are then able to provide a high-probability complexity bound for our method.

Theorem 2.2.1 *Let Assumptions 2.1.1 and 2.1.2 hold. Suppose that Algorithm 2 is applied using a $(\eta, \sigma, P_{\max}, q)$-well-aligned sequence $\{\boldsymbol{P}_j\}$ and a $(\kappa, d_{\max}, p)$-descent sequence $\{\mathcal{D}_j\}$ such that*

$$pq > p_0 := \max\left\{ \frac{\ln(\gamma_{\mathrm{dec}})}{\ln(\gamma_{\mathrm{inc}}^{-1}\gamma_{\mathrm{dec}})}, \frac{\ln(\gamma_{\mathrm{inc}})}{\ln(\gamma_{\mathrm{dec}}^{-1}\gamma_{\mathrm{inc}})} \right\}.$$

For any $\epsilon \in (0,1)$, let N_ϵ be the number of function evaluations necessary to compute an iterate x_J such that $\|\nabla f(x_J)\| \leq \epsilon$. Then,

$$\mathbb{P}\left(N_\epsilon \leq \mathcal{O}\left(m\, \phi\, \epsilon^{-2}\right)\right) \geq 1 - \exp\left[-\frac{(pq - p_0)^2}{4pq} \phi\, \epsilon^{-2} \right], \tag{2.2.9}$$

where

$$\phi = \eta^{-2}\, \sigma^{-2}\, P_{\max}^4\, d_{\max}^8\, \tau^{-2}. \tag{2.2.10}$$

The constants in $\mathcal{O}(\cdot)$ depend on α_0, $f(\boldsymbol{x}_0) - f_{\mathrm{low}}$, γ_{dec}, γ_{inc}, c and L.

As ϵ gets smaller, the bound (2.2.9) becomes more and more certain. For this reason, results such as Theorem 2.2.1 are sometimes referred to as "overwhelming probability" results. Note that complexity bounds in expectation could also be provided.

Columns of $\mathcal{D}_k \setminus \boldsymbol{P}_k$ choice	Identity	Gaussian	s-Hashing	Orthogonal
$[\mathbf{I}_n, -\mathbf{I}_n]$	n^2	n	n	n
Uniform angle PSS	n^3	n	n	n
$[\mathbf{I}_n, -\boldsymbol{e}]$	n^7	n	n	n
Random unit vectors	n	n	n	n

Table 2.1. Summary of evaluation complexity dependency on n for different choices of $\boldsymbol{P}_k$ and $\mathcal{D}_k$ in Algorithm 2.

At first glance, the complexity bound above does not yield any improvement compared to the deterministic guarantee of Theorem 2.1.1. Assessing possible gains from randomness requires a careful analysis of the constant ϕ.

Table 2.1 highlights the various dependencies that can be obtained using several possibilities for $\mathcal{D}_k$ and $\boldsymbol{P}_k$. One sees that the best possible dependency is n, that improves over the best known deterministic dependency in n^2 discussed in Section 2.1.2. In particular, it matches that of (randomized) finite-difference techniques [72, 119].

Figure 2.1 compares several variants of Algorithm 2 on subsets of problems originating from the CUTEst benchmark [34, 69]. Note that the dimensions of those problems pose a particular challenge for direct-search techniques. Indeed, deterministic variants (black and grey curves) struggle to solve a significant fraction of problems within the desired evaluation budget, whereas randomized variants (dashed lines) perform better thanks to their cheap per-iteration cost.

2.2.3 Other subspace methods and extensions

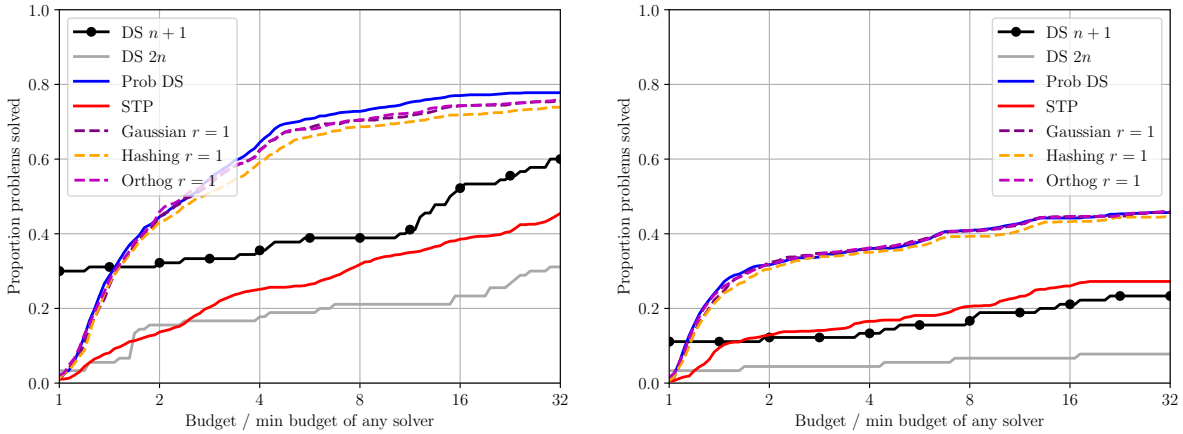
The first use of low-dimensional subspaces in derivative-free optimization with complexity guarantees is arguably due to Nesterov in the context of finite-difference methods [119, 122]. Kozak et al [101, 102] later explored the use of subspaces in finite-difference techniques, going beyond the one-dimensional subspace case of Nesterov. For direct search, Gratton et al. were the first to propose a (one-dimensional) subspace technique with complexity guarantees [77], although it did not make use of the subspace nature of this approach. Model-based derivative-free methods were studied from a subspace perspective by Cartis and Roberts [34], leading to a number of follow-up works [35, 40].

The latest developments in the area revolve around functions with stochastic objectives. Although probabilistic results were established for both direct-search [6, 13, 59, 60] and model-based [17, 37, 39] methods, the use of subspace was relatively underexplored until recently. Dzahini and Wild [62, 63] explored both model-based methods and direct-search techniques, while similar techniques were recently used in the context of zeroth-order/finite-difference schemes [131].

Overall, algorithms based on random subspaces are one of the most prominent lines of research in derivative-free optimization and beyond. Using subspaces not only improves complexity guarantees, but it is also attractive from a practical viewpoint, as it effectively requires less evaluations per iteration while maintaining or even improving the performance of full-space variants [34, 136].

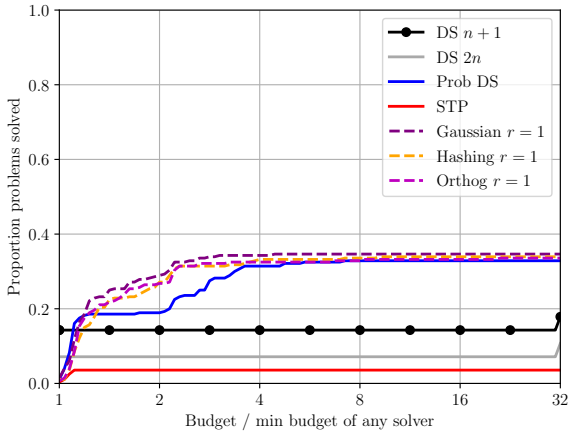
2.3 Positive spanning sets and complexity

In Section 2.2, we discussed how subspace approaches allow to relax the positive spanning set assumption in direct-search methods. In this section, we revert to the standard, deterministic paradigm and consider the problem of generating PSSs with favorable structure and properties. The main chal-

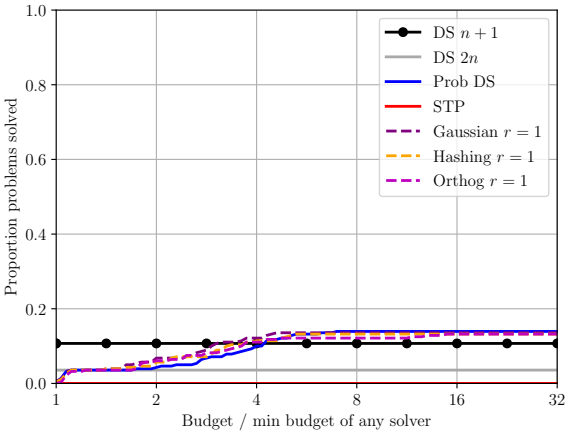


(a) Medium-scale problems, $\tau = 10^{-1}$

(b) Medium-scale problems, $\tau = 10^{-3}$



(c) Large-scale problems, $\tau = 10^{-1}$



(d) Large-scale problems, $\tau = 10^{-3}$

Figure 2.1. Comparison of standard and subspace variants of direct search.

lenge in such a task lies in computing the cosine measure (2.1.2), which is in itself an optimization problem. Section 2.3.1 describes a class of positive spanning sets for which the cosine measure can be provably computed in polynomial time. This class is later used in Section 2.3.2 to build resilient positive spanning sets with a novel definition of cosine measure that can again be computed in polynomial time for certain classes. Section 2.3.3 describes related work on cosine measures.

The results from both Section 2.3.1 and 2.3.2 have been obtained during the PhD thesis of Sébastien Kerleau co-supervised by the author.

2.3.1 Computing the cosine measure easily with OSPBs

Consider a set of vectors $\mathcal{D} \in \mathbb{R}^{n \times m}$, and suppose that this set is a positive spanning set. Hare and Jarry-Bolduc [82] proposed an algorithm that computes the cosine measure of such a set by looking at all possible linear bases contained in the PSS. Although this algorithm computes the cosine measure, the number of linear bases to be considered may be exponential in the problem dimension. In fact, it has recently been shown that computing the cosine measure is a NP-hard problem [93], suggesting that it cannot be done in polynomial time in full generality (unless P=NP).

Such results led to considering special classes of positive spanning sets, for which the computation can be carried in polynomial time.

A positive spanning set $\mathcal{D} \in \mathbb{R}^{n\times m}$ is called a *positive basis* if no proper subset of $\mathcal{D}$ is a PSS. Positive bases possess additional structure compared to PSSs, making them a valuable tool in derivative-free optimization. In particular, it is known that the cardinality of positive bases lies between $n+1$ and $2n$ [7]. Positive bases of size $n+1$ are called *minimal positive bases*, while positive bases of size $2n$ are called *maximal positive bases*. Standard examples for positive bases include the columns of $[\boldsymbol{I}_n \ -\boldsymbol{I}_n]$ (the coordinate vectors and their negatives) and that of $[\boldsymbol{I}_n \ -\boldsymbol{e}]$ (coordinate vectors and the sum of their negatives). For both examples, the cosine measure can be computed explicitly, and this result was used in Table 2.1. However, the result for $[\boldsymbol{I}_n \ -\boldsymbol{e}]$ was not known in the community until very recently [82], and the first proof seem to have been provided in the context of Sébastien Kerleau's work [83].

To obtain such a result, a special class of positive bases was identified, that relies on decomposing positive bases within orthogonal subspaces. To this end, we first observe that Definition 2.1.1 can be generalized to subspaces of $\mathbb{R}^n$. A set $\mathcal{D} \in \mathbb{R}^{n\times m}$ defines a PSS of a subspace $\mathbb{L} \subset \mathbb{R}^n$ provided it generates $\mathbb{L}$ by nonnegative linear combinations. The notion of positive basis of a subspace can be defined accordingly, and leads to the following structure [84].

Definition 2.3.1 *Let $\mathcal{D} \in \mathbb{R}^{n\times m}$ be a positive basis of $\mathbb{R}^n$. If there exist subspaces $\mathbb{L}_1, \ldots, \mathbb{L}_s$ of $\mathbb{R}^n$ and associated minimal positive bases $\mathcal{D}_{\mathbb{L}_1}, \ldots, \mathcal{D}_{\mathbb{L}_s}$ such that*

$$\mathbb{R}^n \ = \ \mathbb{L}_1 \perp \mathbb{L}_2 \perp \cdots \perp \mathbb{L}_s \quad \textit{and} \quad \mathcal{D} \ = \ \mathcal{D}_{\mathbb{L}_1} \cup \mathcal{D}_{\mathbb{L}_2} \cup \cdots \cup \mathcal{D}_{\mathbb{L}_s}, \tag{2.3.1}$$

then $\mathcal{D}$ is called an orthogonally structured positive basis*, or OSPB.*

The aforementioned sets $[\boldsymbol{I}_n \ -\boldsymbol{I}_n]$ and $[\boldsymbol{I}_n \ -\boldsymbol{e}]$ are examples of OSPBs. More generally, any minimal positive basis is an OSPB.

Thanks to the decompositon (2.3.1), computing the cosine measure of an OSPB reduces to computing that of all minimal positive bases involved in its decomposition (at most n). Moreover, for any minimal positive basis of size $\dim(\mathbb{L}_i)+1$, computing the cosine measure is a straightforward calculation [83, Proposition 3.1]. It follows that the cosine measure of an OSPB can be computed in polynomial time [83, Theorem 3.2]. Such a result allows in particular to recover the cosine measures of the sets $[\boldsymbol{I}_n \ -\boldsymbol{I}_n]$ and $[\boldsymbol{I}_n \ -\boldsymbol{e}]$, equal to $\frac{1}{\sqrt{n}}$ and $\frac{1}{\sqrt{n^2+2(n-1)\sqrt{n}}}$, respectively.

2.3.2 Positive k-spanning sets

Using positive bases is valuable in derivative-free optimization, as those are PSSs that are inclusionwise minimal for this property. Yet, using positive bases within an algorithm such Algorithm 1 means that the convergence theory relies on all associated function evaluations being computable. This hypothesis is commonly challenged in applications, either because of hidden constraints that prevents evaluating the function at certain points [96] or because the computing time may be prohibitive for certain calculations [98]. For this reason, researchers have sought more resilient choices of directions than positive spanning sets, leading to the introduction of positive k-spanning sets in the 1980s [111, 112].

Definition 2.3.2 *Given any $k \geq 1$, a set $\mathcal{D}$ of vectors in $\mathbb{R}^n$ is called a* positive k-spanning set *(PkSS) if it is a PSS and any subset $\mathcal{S} \subset \mathcal{D}$ such that $|\mathcal{S}| \geq |\mathcal{D}| - (k-1)$ is also a PSS.*

A PkSS which is minimal for this property is called a positive k-basis.

Both notions of PkSSs and positive k-bases generalize to subspaces of $\mathbb{R}^n$. Note that the definition reduces to that of PSSs and positive bases when $k = 1$.

Although those ideas were introduced in the early 1980s, the use of PkSSs was not investigated in the derivative-free litterature. In fact, the structure of PkSSs itself remains to be understood. For instance, it was quickly established that a PkSS (and thus a positive k-basis) must possess at least $n + 2k - 1$ vectors [111]. However, no upper bound is known for the maximal size of a positive k-basis for general k. Interesting connections between PkSSs and polytopes were pointed out by Woltzlaw [145], revealing that the upper bound lies between $2kn$ and $kn(n + 1)^{k-1}$ [145, Theorem 10.4.2].

Because PkSS were not studied in the context of direct search, there was not equivalent to the cosine measure for those sets. The author introduced such a definition together with Warren Hare, Gabriel Jarry-Bolduc and Sébastien Kerleau [83]. Given $k \geq 1$ and $\mathcal{D} \in \mathbb{R}^{n\times m}$, the *$k$-cosine measure* of $\mathcal{D}$ is

$$\operatorname{cm}_k(\mathcal{D}) := \min_{\substack{\mathcal{S}\subset\mathcal{D} \\ |\mathcal{S}|=|\mathcal{D}|-(k-1)}} \operatorname{cm}(\mathcal{S}). \tag{2.3.2}$$

The k-cosine measure characterizes PkSSs, in that it is positive if and only if the set is a PkSS. Due to its formula, one sees that computing the k-cosine measure of a given set involves computing (1-)cosine measures of certain subsets. Using again OSPBs, we identified classes of PkSSs for which a bound on the k-cosine measure can be found in polynomial time.

Let $\mathcal{D} = \mathcal{D}_{\mathbb{L}_1} \cup \cdots \cup \mathcal{D}_{\mathbb{L}_s} \in \mathbb{R}^{n\times m}$ be an OSPB. Given k real numbers $\beta_1,\ldots,\beta_k$, we consider

$$\mathcal{D}(\beta_1, \ldots, \beta_k) = \bigcup_{i=1}^{k} \beta_i \mathcal{D}, \qquad \text{where} \qquad \beta_i\mathcal{D} = \{\beta_i \boldsymbol{d} \mid \boldsymbol{d} \in \mathcal{D}\}. \tag{2.3.3}$$

If $\beta_1, \ldots, \beta_k$ are nonzero real numbers with distinct absolute values, one can show that $\mathcal{D}(\beta_1, \ldots, \beta_k)$ is a PkSS with

$$\operatorname{cm}_k\left(\mathcal{D}(\beta_1, \ldots, \beta_k)\right) \geq \operatorname{cm}(\mathcal{D}).$$

Such a strategy can be applied to the maximal positive basis $\{\boldsymbol{I}_n\ -\boldsymbol{I}_n\}$.

A more expressive strategy to generate PkSSs consists in applying rotation operators to the elements of $\mathcal{D}$ in a way that preserves the subspace decomposition and does not create duplicates [83, Theorem 4.3]. For any $k \geq 1$, there exists rotations $R_1, \ldots, R_k$ in $\mathbb{R}^n$ such that the set

$$\mathcal{D}(R_1, \ldots, R_k) = \bigcup_{i=1}^{k} R_i(\mathcal{D}), \qquad \text{where} \quad R_i(\mathcal{D}) = \{R_i(\boldsymbol{d}) \mid \boldsymbol{d} \in \mathcal{D}\}, \tag{2.3.4}$$

is a positive k-spanning set. We then have

$$\operatorname{cm}_k\left(\mathcal{D}(R_1, \ldots, R_k)\right) \geq \operatorname{cm}(\mathcal{D}).$$

This rotation-based strategy can be used to obtain PkSSs from the minimal positive basis $\{\boldsymbol{I}_n\ -\boldsymbol{e}\}$. For both strategies, a lower bound on the k-cosine measure can be computed in polynomial time, and thus used to bound the dimension dependencies in complexity results.

To illustrate this observation, we strengthen the complexity results of Algorithm 1 using positive k-spanning sets. We begin by modifying Assumption 2.1.3 as follows.

Assumption 2.3.1 *At every iteration j of Algorithm 1, the polling set $\mathcal{D}_j$ satisfies the following properties:*

(i) $\mathrm{cm}_k(\mathcal{D}_j) \geq \tau_k$ *for some* $\tau_k \in (0,1)$.

(ii) $\forall \boldsymbol{d} \in \mathcal{D}_j,\ d_{\min} \leq \|\boldsymbol{d}\| \leq d_{\max}$.

Using PkSSs makes Algorithm 1 tolerant to stragglers or missing evaluations. In particular, a complexity bound can be obtained while missing up to $k-1$ evaluations at every iteration.

Theorem 2.3.1 *Let Assumptions 2.1.1 and 2.1.2 hold. Suppose that Algorithm 1 is applied with polling sets satisfying Assumption 2.3.1, and that at most* $k-1$ *evaluations fail to complete at every iteration. Then, for any* $\epsilon \in (0,1)$*, the algorithm computes an iterate* x_J *such that* $\|\nabla f(x_J)\| \leq \epsilon$ *in at most*

$$\mathcal{O}\left(\tau_k^{-2}\,\epsilon^{-2}\right) \tag{2.3.5}$$

iterations and

$$\mathcal{O}\left(m\,\tau_k^{-2}\,\epsilon^{-2}\right) \tag{2.3.6}$$

function evaluations.

Provided the PkSSs used in Theorem 2.3.1 are built from OSPBs, a bound τ_k on their cosine measure can be provided. For instance, using rotations of the minimal positive basis $[\boldsymbol{I}_n\ -\boldsymbol{e}]$ yields $m = k(n+1)$ and $\tau_k = \frac{1}{\sqrt{n^2+2(n-1)\sqrt{n}}}$, corresponding to a complexity bound in $\mathcal{O}(k\,n^3)$. Using scaled versions of the maximal positive basis $[\boldsymbol{I}_n\ -\boldsymbol{I}_n]$ gives $m = 2kn$, $\tau_k = \frac{1}{\sqrt{n}}$, so that (2.3.6) is of order $\mathcal{O}(k\,n^2)$. The latter dependency is the best found to date, although it is unclear what the dependencies are for minimal positive k-bases.

2.3.3 Further work on positive spanning sets

Regis [134, 135] studied properties of positive spanning sets by writing the optimality conditions of the optimization problem defining the cosine measure (2.1.2). A number of structural results were derived in Nævdal [115] to characterize the best cosine measures achievable by positive bases of size $n+1$ and $2n$. Hare, Jarry-Bolduc and Planiden [84] introduced the notion of orthogonally structured positive bases (under a different name), and were able to generate positive bases with optimal cosine measure for certain OSPBs.

The thesis of Sébastien Kerleau [97] contains additional results on positive (k-)spanning sets, and their connection with discrete objects such as graphs and polytopes. In particular, using analogies between strongly connected digraphs and positive spanning sets, a decomposition theorem akin to the ear decomposition in graph theory was proposed during the thesis of Sébastien Kerleau [46]. This decomposition revisits an earlier result by Romanowicz [137], but is more practical in that it allows for explicit characterization of families of positive spanning sets of certain cardinality. Characterizing the best tradeoff between the size of a PkSS and its k-cosine measure remains an open problem, even when $k = 1$ [57]. Indeed, this problem bears a close connection with sphere packing and frame theory [128].

2.4 Conclusions and perspectives for Chapter 2

In this section, we presented several complexity results for direct-search algorithms obtained by the author and collaborators. The results from Section 2.2 correspond to a joint work with Lindon Roberts [136], and the associated Python package was incorporated in Meta's Nevergrad platform [132]. The results of Section 2.3 have been partly published in an article with W. Harre, G.

Jarry-Bolduc and S. Kerleau [83]. The complete results are integrally published in the PhD thesis of Sébastien Kerleau [97].

Subspace approaches for derivative-free optimization are currently a growing topic of research, that goes beyond the field of derivative-free optimization as it tackles important scalability issues. A number of methods have been revisited from a subspace perspective in recent years, yet the gain in terms of complexity has been less straightforward than for direct search. Still, practical successes of these approaches suggest that this line of research will keep growing in the future. A relatively underexplored setup is that of subspace methods for constrained derivative-free optimization. A natural continuation of the work conducted by the author and Lindon Roberts on constrained derivative-free optimization [78, 90] would consist in extending the results of Section 2.2 to the constrained setting.

In the context of model-based methods, the use of PSSs is typically replaced with that of fully linear models, that are constructed using either interpolation or regression [42, 43, 44, 45]. As those problems are small-dimensional in general, and convex, they can be solved easily in practice. Nevertheless, the cost of building those models can be harder to quantify than in direct search, where the steps are somewhat more explicit. Still, assessing the quality of linear models remains a topic of interest in the derivative-free optimization community [113, 130]. Since positive spanning sets can be used to construct linear models in model-based derivative-free optimization [45], our results offer a natural path investigate the quality of these linear models (in the spirit of Section 2.3.1) or the construction of such models using PkSSs (described in Section 2.3.2).

Chapter 3

Complexity of conjugate gradient methods

This chapter is concerned with first-order methods, that rely on gradient and function evaluations to perform optimization. A number of first-order schemes have been thoroughly analyzed from a complexity viewpoint in convex and strongly convex settings, thereby allowing to classify algorithms according to their complexity guarantees. By contrast, in a general nonconvex setting, the basic complexity bound of gradient-based algorithms has proven difficult to beat, surprisingly setting all methods equal from a complexity perspective. The author's work has focused on the class of *conjugate gradient* methods, which can exhibit good numerical behavior when applied to numerical problems. By introducing tweaks to the original algorithm, one can equip them with complexity bounds that reflect their original empirical behavior.

Section 3.1 introduces the canonical algorithm within that class of methods, *gradient descent*, as well as the main complexity results for that algorithm. Section 3.2 then presents results obtained by the author during his postdoctoral studies on conjugate gradient for the special case of nonconvex quadratic functions. Section 3.3 returns to the general nonconvex setting, and describes results obtained by the author together with his former intern Rémi Chan--Renous-Legoubin on nonlinear conjugate gradient methods. Section 3.4 discusses short-term perspectives on this work.

3.1 Gradient descent and fundamental complexity results

Throughout this section, we are concerned with the optimization problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\boldsymbol{x}), \tag{3.1.1}$$

where f is a continuously differentiable function with Lipschitz continuous gradient. This setting is arguably one of the most classical in mathematical optimization, as this assumption holds for a number of convex (linear regression, logistic regression) and nonconvex (sigmoid regression) problems. It is also the simplest paradigm under which one can derive complexity guarantees for gradient-based techniques [118, 28].

Section 3.1.1 presents a gradient-based algorithm that will be used as prototypical algorithm for the rest of this chapter. Complexity guarantees for this method are discussed in Section 3.1.2, and other results in the literature are surveyed in Section 3.1.3.

3.1.1 A basic gradient algorithm with line search

Gradient descent is the canonical algorithm for optimizing smooth, nonlinear functions. It consists in moving along the negative gradient direction at each iteration. A crucial component of this method lies in the length of such a move, also called stepsize, akin to the stepsize in direct-search schemes.

Algorithm 3: Gradient descent algorithm.

Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma_{\text{dec}} \in (0,1)$, $c \in (0,1)$.
for $k = 0, 1, \ldots$ **do**

1. Compute the gradient $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$.
2. Find the largest stepsize $\alpha \in \{ \gamma_{\text{dec}}^j \alpha_0 \mid j \in \mathbb{N} \}$ such that
$$f(\boldsymbol{x}_k - \alpha \boldsymbol{g}_k) < f(\boldsymbol{x}_k) - c\,\alpha \|\boldsymbol{g}_k\|^2 \tag{3.1.2}$$
3. Set $\alpha_k = \alpha$ and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)$.

end

Algorithm 3 relies on a backtracking (also known as Armijo) line search to compute a suitable stepsize for moving along the negative gradient direction. Although textbook analyses for gradient descent consider a constant or a decreasing stepsize, it is also possible to derive complexity guarantees for this method using line-search schemes [149, 36]. We present such a guarantee in the next section.

3.1.2 Complexity analysis of gradient descent

We consider the same setup than in Chapter 2, i.e. the minimization of a $\mathcal{C}^{1,1}$ function. A complexity result was obtained by Cartis, Sampaio and Toint [36] for gradient-based method using line search. The proof for Algorithm 3 can be found in a number of textbooks, such as the recent monograph of Wright and Recht [149].

Theorem 3.1.1 *Suppose that Algorithm 3 is applied to problem (3.1.1) under Assumptions 2.1.2 and 2.1.1. Then, the method computes an iterate $\boldsymbol{x}_J$ such that $\|\nabla f(\boldsymbol{x}_J)\| \leq \epsilon$ in at most*

$$\mathcal{O}(\epsilon^{-2}) \tag{3.1.3}$$

iterations or, equivalently gradient evaluations, where the constant in $\mathcal{O}(\cdot)$ depends on L, $f(\boldsymbol{x}_0)$, f_{low}, c, α_0 and γ_{dec}. The associated cost in terms of function evaluations is at most

$$\mathcal{O}(\epsilon^{-2}), \tag{3.1.4}$$

where the constant in (3.1.4) differs from that of (3.1.3) by a factor logarithmic in α_0, c, L and γ_{dec}.

The dependency ϵ^{-2} was shown to be essentially sharp [28], through the introduction of an example for which Algorithm 3 takes at least $\mathcal{O}(\epsilon^{-2+\delta})$ evaluations to reach an ϵ-stationary point for any $\delta > 0$. This small-dimensional example was later refined to obtain an $\mathcal{O}(\epsilon^{-2})$ bound [33]. Interestingly, larger-dimensional examples, for which the problem dimension is defined according to the complexity bound, have also been proposed [26].

3.1.3 Extensions to other gradient-based schemes

The use of gradient-related directions [149] is a classical paradigm under which complexity guarantees for gradient descent are preserved (up to constant factors). Rather than using the negative gradient as a search direction, one considers a vector $\boldsymbol{d}_k$ such that

$$\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k \leq -\sigma\|\boldsymbol{d}_k\|\,\|\boldsymbol{g}_k\| \quad \text{and} \quad \|\boldsymbol{d}_k\| \leq \kappa\|\boldsymbol{g}_k\|. \tag{3.1.5}$$

These conditions, together with a modified line-search condition (3.1.2) (involving $\|\boldsymbol{d}_k\|^2$ or $\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{g}_k$ rather than $\|\boldsymbol{g}_k\|^2$) then lead to complexity guarantees of the same order than that of Theorem 3.1.1 in terms of dependencies of ϵ. Moreover, a nonmonotone strategy, that consists in measuring decreases with respect to the largest function value encountered over the past few iterations, can be used with similar complexity guarantees than that described in the previous section [36].

Besides gradient descent, trust-region methods with first-order guarantees (not necessarily relying on exact second-order information) also reach an ϵ-first order stationary point in at most $\mathcal{O}(\epsilon^{-2})$ iterations [79]. It is known that this dependency on ϵ is tight for continuously differentiable functions with Lipschitz continuous derivatives [27, 33].

3.2 Linear conjugate gradient and nonconvex quadratics

In this section, we consider a specific class of continuously differentiable problems, namely quadratic problems of the form

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, q(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{x}, \tag{3.2.1}$$

where $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ is a symmetric matrix, and $\boldsymbol{g} \in \mathbb{R}^n$. Such problems typically arise as subproblems in second-order optimization methods, such as those to be described in Chapter 4.

When the matrix $\boldsymbol{H}$ is positive semidefinite (resp. positive definite), problem (3.2.1) is a convex (resp. strongly convex) optimization problem, and it can be solved efficiently using first-order methods. Among those techniques, the conjugate gradient method [89] has remained one of the most popular approaches for tackling such problems, that is guaranteed to converge in n iterations under exact arithmetic. In addition, a convergence rate can be obtained for conjugate gradient on strongly convex quadratics [123, Theorem 5.5 and equation (5.36)]. This rate matches that of accelerated techniques such as Nesterov's accelerated gradient and Polyak's heavy ball method [148].

When the matrix $\boldsymbol{H}$ is not positive semidefinite, problem (3.2.1) is unbounded, as the function q decreases indefinitely along *negative curvature directions*. More precisely, a negative curvature direction for problem (3.2.1) is a (nonzero) vector $\boldsymbol{d} \in \mathbb{R}^n$ such that

$$\boldsymbol{d}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{d} < 0 \quad \text{and} \quad \boldsymbol{d}^{\mathrm{T}}\boldsymbol{g} \leq 0. \tag{3.2.2}$$

If such a direction exists, it follows that the problem is unbounded. When problem (3.2.1) arises as a subproblem of a nonlinear optimization method, the direction of negative curvature can be used to define a step for this algorithm [41].

Given a quadratic problem (3.2.1), we thus seek an algorithm that either outputs an approximate solution of the problem if it exists, or identifies nonconvexity by exhibiting a direction of negative curvature.

3.2.1 Gradient descent and Krylov methods

A first possibility to tackle the problem mentioned at the end of the previous section consists in applying gradient descent to the nonconvex quadratic problem (3.2.1). When $\boldsymbol{g} = \boldsymbol{0}$, gradient descent resembles the power method, a well-known strategy for approximating extreme eigenvalues in linear algebra [68]. In the case of a strongly convex quadratic function, gradient descent is guaranteed to converge to an approximate solution in a number of iterations that is logarithmic in the optimality tolerance [120]. This bound can even be improved using accelerated techniques such as Nesterov's accelerated gradient or Polyak's heavy-ball method [55]. In addition, Krylov methods such aslinear conjugate gradient exhibit acceleration properties, in the sense that they converge at an accelerated rate [94].

When the quadratic is nonconvex, two perspectives can be adopted on problem (3.2.1). On consists in regularizing the objective using a trust-region constraint or a cubic regularization term. In both cases, an analysis of gradient descent and Krylov subspace methods can then be conducted [22, 23]. In order to guarantee that the method does converge to a minimum, adding randomness may be required, leading to convergence rates that hold with high probability [24].

The other perspective on problem (3.2.1) consists in estimating how fast an algorithm can escape saddle points (i.e. first-order stationary points that are not local minima). Interest for such results rose in the late 2010s following a breakthrough result by Lee et al. [106, 105], that showed that gradient descent almost never (in a probability sense) converges towards a saddle point at which the Hessian matrix is indefinite. Nevertheless, the method can require exponential time to escape the vicinity of such a saddle point [58]. Randomness again alleviates this issue, and convergence rates for converging to a second-order stationary point have been obtained for gradient descent [91], accelerated gradient [92] as well as heavy ball [124]. A core idea consists in looking near the saddle point, where the function resembles a nonconvex quadratic function.

3.2.2 Revisiting linear conjugate gradient

In order to tackle the original problem of interest (3.2.1), which may potentially be nonconvex, we considered two uses of conjugate gradient. On one hand, we aim at computing an approximate solution of the problem if there exists one. On the other hand, we would like to use linear conjugate gradient to detect negative curvature if it exists, in line with practical behavior.

Algorithm 4 states the basic conjugate gradient method [123, Chapter 5] applied to a modified version of problem (3.2.1), namely

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, q(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\bar{\boldsymbol{H}}\boldsymbol{x} + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{x}, \tag{3.2.3}$$

where $\bar{\boldsymbol{H}}$ is a shifted version of $\boldsymbol{H}$. In exact arithmetic, this algorithm converges in n iterations to a solution when the quadratic is strongly convex. Otherwise, the method can stop when detecting a negative curvature direction. Together with Michael O'Neill and Stephen Wright, the author quantified this phenomenon by considering a regularized version of problem (3.2.1).

Theorem 3.2.1 *([138, Corollary 3]) Suppose that conjugate gradient is applied to the modified problem (3.2.3) with $\bar{\boldsymbol{H}} = \boldsymbol{H} + \epsilon\,\boldsymbol{I}_n$ and $\bar{\boldsymbol{g}}$ a random vector uniformly drawn in the unit sphere. Let $p \in (0, 1]$ and*

$$\bar{J} = \min\left\{n, 1 + \left\lceil \ln(2.75\,n/(1-p)^2)\|H\|^{1/2}\,\epsilon^{-1/2}\right\rceil\right\}. \tag{3.2.4}$$

Then, with probability p, one of the two events below occur:

Algorithm 4: Conjugate gradient algorithm for problem (3.2.1).

Initialization: $\boldsymbol{r}_0 = \boldsymbol{g}$, $\boldsymbol{p}_0 = -\boldsymbol{r}_0$, $\boldsymbol{y}_0 = \boldsymbol{0}$, $j = 0$.
while $\boldsymbol{p}_j^{\mathrm{T}} \bar{H} \boldsymbol{p}_j > 0$ *and* $\|\boldsymbol{r}_j\| > 0$ **do**

1. Set $\alpha_j = \frac{\|\boldsymbol{r}_j\|^2}{\boldsymbol{p}_j^{\mathrm{T}} \bar{\boldsymbol{H}} \boldsymbol{p}_j}$
2. Compute $\boldsymbol{y}_{j+1} = \boldsymbol{y}_j + \alpha_j \boldsymbol{p}_j$
3. Compute $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j + \alpha_j \bar{\boldsymbol{H}} \boldsymbol{p}_j$
4. Compute $\beta_{j+1} = \frac{\|\boldsymbol{r}_{j+1}\|^2}{\|\boldsymbol{r}_j\|^2}$
5. Set $\boldsymbol{p}_{j+1} = -\boldsymbol{r}_{j+1} + \beta_{j+1} \boldsymbol{p}_j$.
6. Set $j = j + 1$.

end

(i) Algorithm 4 computes $\boldsymbol{p}_j$ such that $\boldsymbol{p}_j^{\mathrm{T}} \bar{H} \boldsymbol{p}_j < 0$ in $j \leq \bar{J}$ iterations, or

(ii) Algorithm 4 does not terminate before iteration $\bar{J}$, in which case it provides a certificate that $\bar{\boldsymbol{H}} \succeq -\frac{\epsilon}{2} \boldsymbol{I}_n$ with probability p.

In addition, we proposed a version of conjugate gradient, called Capped Conjugate Gradient (or Capped CG), that uses extra checks and stopping criteria to detect nonconvexity in a deterministic fashion. Our inspiration stemmed from the accelerated gradient technique of Carmon et al. [25], and consists in appropriately regularizing the objective function to guarantee sufficient curvature when needed.

As a result, the number of iterations of Algorithm 5 does not exceed $\min\left\{n, \tilde{\mathcal{O}}(\epsilon^{-1/2})\right\}$ [138, Lemma 1]. This complexity is on par with that of accelerated methods. We again emphasize that this result applies to nonconvex quadratics: either the method computes an approximate solution to the linear system $\bar{\boldsymbol{H}} \boldsymbol{y} = -\boldsymbol{g}$, or it outputs a direction of small curvature for $\bar{\boldsymbol{H}}$, i.e. a direction of negative curvature for $\boldsymbol{H}$.

3.2.3 Applications and extensions

Together with Michael J. O'Neill and Stephen J. Wright, the author proposed a Newton-line search algorithm based on using the capped CG routine [138], that achieves the best known iteration and evaluation complexity among second-order methods. We will discuss a variant of this algorithm in Chapter 4. A subsequent line of work due to Yang Liu and Fred Roosta, based on minimum residual methods such as conjugate residuals or MINRES, demonstrated that such methods were competitive with conjugate gradient in terms of detecting negative curvature [110].

A number of extensions of the capped CG algorithm have been proposed in constrained optimization settings. Yue Xie and Stephen J. Wright have use capped CG within algorithms for nonconvex optimization with bound constraints [151] and nonlinear equality constraints [150]. Chuan He and Zhaosong Lu, along with collaborators, investigated extension of Newton-Capped CG to conic and

Algorithm 5: Capped Conjugate Gradient

Inputs: Symmetric matrix $\boldsymbol{H} \in \mathbb{R}^{n\times n}$; vector $\boldsymbol{g} \neq 0$; damping parameter $\epsilon \in (0,1)$; desired relative accuracy $\zeta \in (0,1)$.
Optional input: scalar M (set to 0 if not provided).
Outputs: d_type, $\boldsymbol{d}$.
Secondary outputs: final values of M, κ, $\hat{\zeta}$, τ, and T.
Set

$$\bar{\boldsymbol{H}} := \boldsymbol{H} + 2\epsilon \boldsymbol{I}_n, \quad \kappa := \frac{M+2\epsilon}{\epsilon}, \quad \hat{\zeta} := \frac{\zeta}{3\kappa}, \quad \tau := \frac{\sqrt{\kappa}}{\sqrt{\kappa}+1}, \quad T := \frac{4\kappa^4}{(1-\sqrt{\tau})^2}.$$

$\boldsymbol{y}_0 \leftarrow 0$, $\boldsymbol{r}_0 \leftarrow \boldsymbol{g}$, $\boldsymbol{p}_0 \leftarrow -\boldsymbol{g}$, $j \leftarrow 0$.
If $\boldsymbol{p}_0^T \bar{\boldsymbol{H}} \boldsymbol{p}_0 < \epsilon \|\boldsymbol{p}_0\|^2$, set $\boldsymbol{d} = \boldsymbol{p}_0$ and terminate with d_type=NC.
Elseif $\|\boldsymbol{H}\boldsymbol{p}_0\| > M\|\boldsymbol{p}_0\|$ set $M \leftarrow \|\boldsymbol{H}\boldsymbol{p}_0\|/\|\boldsymbol{p}_0\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly.
while *True* **do**

$\alpha_j \leftarrow \boldsymbol{r}_j^T \boldsymbol{r}_j / \boldsymbol{p}_j^T \bar{\boldsymbol{H}} \boldsymbol{p}_j$
$\boldsymbol{y}_{j+1} \leftarrow \boldsymbol{y}_j + \alpha_j \boldsymbol{p}_j$
$\boldsymbol{r}_{j+1} \leftarrow \boldsymbol{r}_j + \alpha_j \bar{\boldsymbol{H}} \boldsymbol{p}_j$
$\beta_{j+1} \leftarrow (\boldsymbol{r}_{j+1}^T \boldsymbol{r}_{j+1})/(\boldsymbol{r}_j^T \boldsymbol{r}_j)$
$\boldsymbol{p}_{j+1} \leftarrow -\boldsymbol{r}_{j+1} + \beta_{j+1} \boldsymbol{p}_j$
$j \leftarrow j+1$

If $\|\boldsymbol{H}\boldsymbol{p}_j\| > M\|\boldsymbol{p}_j\|$, set $M \leftarrow \|\boldsymbol{H}\boldsymbol{p}_j\|/\|\boldsymbol{p}_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly.
If $\|\boldsymbol{H}\boldsymbol{y}_j\| > M\|\boldsymbol{y}_j\|$, set $M \leftarrow \|\boldsymbol{H}\boldsymbol{y}_j\|/\|\boldsymbol{y}_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly.
If $\|\boldsymbol{H}\boldsymbol{r}_j\| > M\|\boldsymbol{r}_j\|$, set $M \leftarrow \|\boldsymbol{H}\boldsymbol{r}_j\|/\|\boldsymbol{r}_j\|$ and update $\kappa, \hat{\zeta}, \tau, T$ accordingly.

If $\|\boldsymbol{r}_j\| \leq \hat{\zeta}\|\boldsymbol{r}_0\|$, set $\boldsymbol{d} \leftarrow \boldsymbol{y}_j$ and terminate with d_type=SOL
Elseif $\boldsymbol{y}_j^T \bar{\boldsymbol{H}} \boldsymbol{y}_j < \epsilon \|\boldsymbol{y}_j\|^2$, set $\boldsymbol{d} \leftarrow \boldsymbol{y}_j$ and terminate with d_type=NC
Elseif $\boldsymbol{p}_j^T \bar{\boldsymbol{H}} \boldsymbol{p}_j < \epsilon \|\boldsymbol{p}_j\|^2$, set $\boldsymbol{d} \leftarrow p_j$ and terminate with d_type=NC
Elseif $\|\boldsymbol{r}_j\| > \sqrt{T}\tau^{j/2}\|\boldsymbol{r}_0\|$

- Compute $\alpha_j, \boldsymbol{y}_{j+1}$ as in the main loop above
- Find $i \in \{0, \ldots, j-1\}$ such that

$$\frac{(\boldsymbol{y}_{j+1} - \boldsymbol{y}_i)^T \bar{\boldsymbol{H}} (\boldsymbol{y}_{j+1} - \boldsymbol{y}_i)}{\|\boldsymbol{y}_{j+1} - \boldsymbol{y}_i\|^2} < \epsilon. \tag{3.2.5}$$

- Set $\boldsymbol{d} \leftarrow \boldsymbol{y}_{j+1} - \boldsymbol{y}_i$ and terminate with d_type=NC.

end

nonlinear constraints [85, 87, 88]. An extension to functions with Hölder-continuous Hessians was also recently proposed [86].

Finally, a Newton-type method for nonconvex least squares problems was investigated during the PhD thesis of Iskander Legheraba [107]. This method relies on capped CG to solve a quadratic subproblem at every iteration.

3.3 Restarting nonlinear conjugate gradient

In this section, we come back to the general nonlinear nonconvex optimization setting and consider problem (3.1.1), which we aim at tackling using line-search gradient-based methods. Conventional wisdom in nonlinear optimization suggests that a number of techniques can be more efficient than gradient descent in practice [123]. In particular, nonlinear conjugate gradient methods remain an important topic of investigation in large-scale optimization, with demonstrated practical interest [80, 129].

Interestingly, a work by Carmon et al. [25] found that a very basic implementation of nonlinear conjugate gradient was capable of outperforming gradient descent as well as accelerated variants designed for complexity purposes on robust regression tasks. This observation was the starting point for studying nonlinear conjugate gradient techniques from a complexity perspective, through the master internship of Rémi Chan--Renous-Legoubin at Université Paris Dauphine-PSL.

Section 3.3.1 describes the algorithm of interest, in connection with the gradient descent algorithm studied in Section 3.1. Section 3.3.2 presents results that were obtained for nonlinear conjugate variants and later published in *EURO Journal on Computational Optimization* [38]. Section 3.3.3 discusses extensions of this approach, following in particular recent developments in the noisy setting.

3.3.1 A restarting framework for line-search methods

As explained in Section 3.1, deriving complexity results for gradient-based methods in a line-search framework requires control over the two main components of a line-search scheme. On the one hand, the search direction must be a sufficient descent direction, i.e. it must make a sufficiently negative inner product with the negative gradient. On the other hand, the norm of this direction should be related to that of the gradient.

These observations were the motivation behind introducing Algorithm 6. At every iteration, the method is allowed to pick a direction according to the current gradient and possibly information from previous iterations. Before performing a line search along this direction, the algorithm checks if that direction is in sufficient agreement with the gradient. If not, then this direction is replaced by the negative gradient and the corresponding iteration is called a *restarted* iteration. For such an iteration, we thus have $\boldsymbol{d}_k = -\boldsymbol{g}_k$ and

$$\boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{g}_k = -\|\boldsymbol{g}_k\|^2 \qquad \text{and} \qquad \|\boldsymbol{d}_k\| = \|\boldsymbol{g}_k\|. \tag{3.3.3}$$

For a non-restarted iteration, the converse of condition (3.3.1) holds, i.e.

$$\boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{g}_k \leq -\kappa \|\boldsymbol{g}_k\|^{1+p} \qquad \text{and} \qquad \|\boldsymbol{d}_k\| \leq \kappa^{-1} \|\boldsymbol{g}_k\|^{\frac{1+p}{2}}. \tag{3.3.4}$$

Using $p = 1$ in (3.3.4) yields a condition that resembles that for line-search methods based on descent directions [36]. Still, the purpose of introducing the parameter p was to go beyond the classical condition, in order to obtain a better complexity for the method.

Algorithm 6: Line-search algorithm with restarting condition.

Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma_{\text{dec}} \in (0,1)$, $c \in (0,1)$, $p \in [0,1]$, $\kappa \in (0,1)$.
for $k = 0, 1, \ldots$ **do**

1. Compute the gradient $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$.
2. Compute a gradient-based direction $\boldsymbol{d}_k$ using $\boldsymbol{g}_k$ and possibly past information.
3. If the restarting condition holds, i.e. if
$$\boldsymbol{d}_k^{\mathrm{T}} \boldsymbol{g}_k \geq -\kappa \|\boldsymbol{g}_k\|^{1+p} \qquad \text{or} \qquad \|\boldsymbol{d}_k\| \geq \kappa^{-1} \|\boldsymbol{g}_k\|^{\frac{1+p}{2}}, \tag{3.3.1}$$
set $\boldsymbol{d}_k = -\boldsymbol{g}_k$.
4. Find the largest stepsize $\alpha \in \{ \gamma_{\text{dec}}^j \alpha_0 \mid j \in \mathbb{N} \}$ such that
$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) < f(\boldsymbol{x}_k) + c\, \alpha \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{d}_k \tag{3.3.2}$$
5. Compute a steplength $\alpha_k > 0$.
6. Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)$.

end

3.3.2 Restarted nonlinear conjugate gradient

Together with Rémi Chan–Renous-Legoubin [38], the author focused on a variant of Algorithm 6 based on nonlinear conjugate gradient methods [81]. At every iteration $k \geq 1$, we first compute $\boldsymbol{d}_k$ as

$$\boldsymbol{d}_k = -\nabla f(\boldsymbol{x}_k) + \beta_k \boldsymbol{d}_{k-1}, \tag{3.3.5}$$

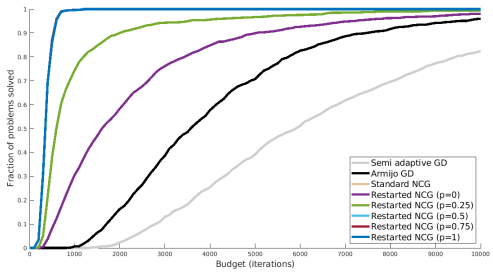
where β_k is a coefficient that depends on the particular method at hand (note that we set $\boldsymbol{d}_0 = -\nabla f(\boldsymbol{x}_0)$). In our experiments, and in agreement with previous investigation [25], we set β_k using the PRP+ (Polak-Ribière-Polyak+) rule, i.e.

$$\beta_k = \max \left[\frac{\nabla f(\boldsymbol{x}_k)^{\mathrm{T}} (\nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1}))}{\|\nabla f(\boldsymbol{x}_{k-1})\|^2}, 0 \right]. \tag{3.3.6}$$

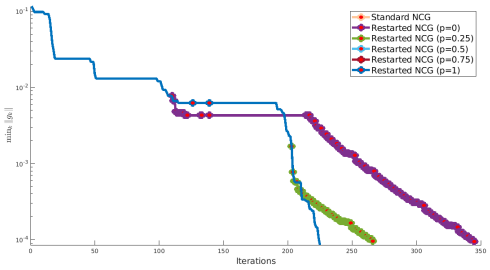
Note that this parameter is known to guarantee global convergence [54, 67], but that no complexity results were established prior to this work. Our goal in obtaining such a complexity bound was to identify the possible benefit of taking nonlinear CG iterations, which can be assessed thanks to condition (3.3.4).

Theorem 3.3.1 *Suppose that Assumptions 2.1.1 and 2.1.2 hold. Suppose that Algorithm 6 is applied using the formula (3.3.5) to compute $\boldsymbol{d}_k$ without restarts. Then, the method reaches $\boldsymbol{x}_K$ such that $\|\nabla f(\boldsymbol{x}_k)\| \leq \epsilon$ in at most*

$$\mathcal{O}(\epsilon^{-2}) + \mathcal{O}\left(\epsilon^{-(1+p)}\right), \tag{3.3.7}$$



(a) Comparison between restarted methods and other first-order methods. The curves corresponding to Restarted NCG with $p \in \{0.5, 0.75, 1\}$ overlap with that of standard nonlinear CG.



(b) A representative instance of the robust regression problem of interest. The curves corresponding to Restarted NCG with $p \in \{0.5, 0.75, 1\}$ overlap with that of standard nonlinear CG, and all have the same 2 restarted iterations (blue circles with red filing). Restarted NCG($p = 0.25$) variant had 66 restarted iterations, while Restarted NCG($p = 0$) had 148 restarted iterations.

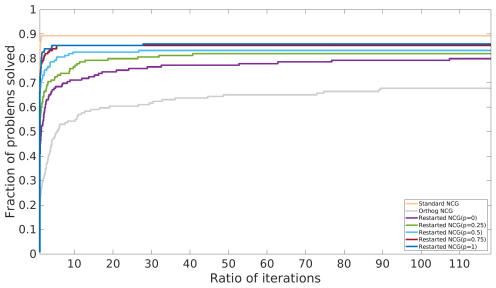
Figure 3.1. Comparison of restarted nonlinear CG variants with non-restarted nonlinear CG, gradient descent and a semi-adaptive variant [25]. All restarting variants use $\kappa = 10^{-2}$ and the PRP+ update (3.3.6).

iterations, where the first term is a bound on the number of restarted iterations and the second is a bound on the number of non-restarted iterations.

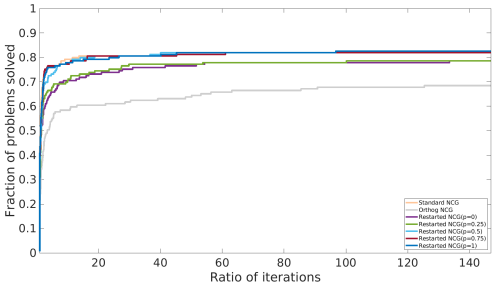
Due to the possibility of restarts, the bound (3.3.7) does not improve over the bound of gradient descent from Section 3.1. Indeed, in the worst case, one may have to restart at every iteration, in which case the algorithm is equivalent to gradient descent.

Nevertheless, if the number of restarted iterations is relatively low compared to the total number of iterations, one may consider the second term as being more illustrative of the algorithmic behavior. Using both the robust statistics problem from Carmon et al. [25] and problems from the CUTEst collection [69], we validated this hypothesis empirically. In practice, we found that using $p \geq 0.5$ led to very similar performance compared to standard nonlinear CG on a benchmark of robust regression instances generated using Bernoulli noise and Tukey biweight loss [25], in that the method almost never restarted. As shown in Figure 3.1, the performance of the nonlinear CG variants is better than that of gradient descent techniques, but it deteriorates as one decreases the value of p, even though using a small value for p seemingly improves part of the complexity bound in (3.3.7). In fact, decreasing p also increases the number of restarts, and the method behaves more like a gradient descent scheme than a nonlinear conjugate gradient one.

Overall, our study revealed that checks can be added to a nonlinear conjugate gradient technique in order to equip it with complexity guarantees without compromising its practical performance. In our setting, choosing the value of p was critical, and values close to 1 led to the best performance on our targeted problem. Our experiments on CUTEst show a small deterioration in performance upon adding the restart condition (see Figure 3.2), even though the variants with $p \geq 0.5$ still perform close to the vanilla method. Interestingly, for the Fletcher-Reeves nonlinear CG variant, which is known for being more amenable to theoretical analyses [81], the restarting condition seems to have little impact on the performance for sufficiently large p.



(a) PRP+ formula (3.3.6).



(b) Fletcher-Reeves formula $\beta_k = \frac{\|\nabla f(\boldsymbol{x}_k)\|^2}{\|\nabla f(\boldsymbol{x}_{k-1})\|^2}$.

Figure 3.2. Comparison between standard nonlinear CG [123], orthogonal nonlinear CG [95], and several restarted nonlinear CG variants, on CUTEst test problems.

3.3.3 Extensions of the restarted framework

Algorithm 6 is an algorithmic framework, that can be combined with any gradient-type technique (and possibly other line-search strategies). In particular, quasi-Newton algorithms, that form one of the most efficient methods in nonlinear optimization, can be controlled with a restarted condition in the same spirit as nonlinear conjugate gradient methods. In a report recently submitted with Albert Berahas and Michael O'Neill [12], the author investigates the combination of L-BFGS [109] with the restarted condition (3.3.1). Although this particular condition seem less appropriate for L-BFGS techniques, we again observe that parameters (p, κ) can be defined such that the condition is not often triggered. Interestingly, the best value of p is less than 1, leading to a low number of restarts and, as a result, to a favorable balance between restarted and non-restarted iterations for our complexity bound.

This recent study also investigates noisy problems, in which function and gradient values are replaced by noisy estimates. By enforcing control on the noise similarly to existing literature [11], we obtain convergence results to a neighborhood of a stationary point.

3.4 Conclusion and perspectives for Chapter 3

Deriving complexity bounds for gradient descent on a nonconvex optimization problem is one of the most classical components of modern-day optimization courses. Although the analysis is typically performed using fixed stepsizes, using a line search leads to similar overall guarantees while quantifying the cost of not knowing problem-dependent quantities such as the Lipschitz constant for the gradient. To the best of our knowledge, despite other line-search procedures leading to better practical performance [123], the connection between line search and complexity guarantees remains underexplored. In particular, it is unclear whether more elaborate line searches and direction choices can improve over the classical gradient descent method with Armijo line search in terms of complexity.

The simplest class of nonlinear optimization problems is that of quadratic optimization problems. In nonconvex optimization, nonconvex quadratic problems arise naturally in algorithms for more general problems, and must be tackled efficiently to compute appropriate steps for the overall problem. The author's research has focused on tackling this task using linear conjugate gradient, arguably the method of choice for solving strongly convex quadratic problems. Regularizing the quadratic term

proved instrumental to obtain complexity results. Those results highlighted the method's ability to either solve the problem to sufficient accuracy or detect sufficient nonconvexity when present. Numerous linear algebra and iterative techniques can be used to tackle strong convex quadratic problems while possessing optimal or near-optimal complexity guarantees [55]. However, not all of them have been endowed with guarantees in a nonconvex setting, and investigating such an extension is a natural continuation of the work conducted with Michael J. O'Neill and Stephen J. Wright [125].

Classical numerical optimization techniques fail to improve over gradient descent, despite overperforming gradient descent on typical benchmarks. The restarting framework introduced in Section 3.3 is one attempt as connecting practice and theory, by adding checks on the directions that are used to update the iterate, and limiting those that deviate significantly from gradient descent steps. For nonlinear conjugate gradient, such an approach gave rise to a method close to textbook nonlinear conjugate gradient but with complexity guarantees. In addition to revisiting other nonlinear conjugate gradient and quasi-Newton schemes with a restarting perspective, one could also go beyond these methods to tackle noisy and stochastic settings. To this end, one could rely upon *objective-function-free algorithms* for nonconvex optimization [73, 74]. Extending the restarting framework of Section 3.3 would remove the need for (possibly noisy) function estimates without compromising complexity guarantees.

Chapter 4

Complexity of trust-region Newton methods

Newton-type methods are based on incorporating second-order derivative information into optimization algorithms so as to improve over first-order schemes. This seemingly appealing property does not directly translate into improved complexity results, as basic variants of Newton's method exhibit the same worst-case complexity than gradient descent techniques [28]. Still, a number of algorithms have been proposed that satisfy optimal complexity bounds provided the function to be minimized is sufficiently smooth. In this chapter, we will again consider the problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\boldsymbol{x}), \tag{4.0.1}$$

but will now assume that the function f is twice continuously differentiable with a Lipschitz continuous Hessian. Under this assumption, we seek algorithms that can be applied to problem (4.0.1) and compute approximate second-order stationary points in the sense of (1.1.3).

In this chapter, we illustrate the author's work on Newton-type methods with complexity guarantees through the lens of trust-region methods. Section 4.1 discusses the challenges in equipping Newton's method with good complexity results. Section 4.2 shows how a standard globalization technique of Newton's method (trust regions) can be endowed with best-known complexity bounds for second-order techniques. Finally, Section 4.3 improves the complexity of Newton trust-region schemes when the problem at hand has favorable landscape.

4.1 Newton's method and an issue with complexity

In its most elementary form, Newton's method is not well defined, and globalization techniques (line search, trust region, regularization) are required to make it globally convergent. Still, it is known to exhibit fast (quadratic) local convergence rates in the vicinity of minimizers around which the function is locally strongly convex [20]. From a complexity point of view, Newton's method does not possess better guarantees than gradient descent on nonconvex $\mathcal{C}^{1,1}$ problems [28]. However, for $\mathcal{C}^{2,2}$ functions, a more refined analysis can be performed and improvement over first-order complexity bounds can be established.

Section 4.1.1 presents known complexity results for Newton-trust region schemes, that leverage second-order derivative information. Section 4.2.1 explains how the first method can be modified to

obtain best-known complexity results for the class of $\mathcal{C}^{2,2}$ nonconvex objective functions. Section 4.3.2 discusses strict saddle functions, a particular subclass of nonconvex problems for which even a basic trust-region method gets improved complexity guarantees.

4.1.1 A classical Newton-trust region technique

Algorithm 7 presents a standard trust-region algorithm using a second-order model [41]. The magnitude of Newton steps is controlled by enforcing a trust-region constraint while computing a step, which is written as a quadratic optimization problem over a ball (4.1.1). At each iteration, depending on the agreement between the model and the true objective, the step is either accepted or rejected, and the trust-region radius Δ_k is either expanded or shrunk (though more sophisticated rules exist [41, 152]).

Algorithm 7: Basic Newton-trust region algorithm.

Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\Delta_0 > 0$, $\Delta_{\max} > \Delta_0$, $c \in (0, 1)$.
for $k = 0, 1, \ldots$ **do**

1. Compute a tentative step by solving the trust-region subproblem

$$\underset{\substack{\boldsymbol{s}\in\mathbb{R}^n \\ \|\boldsymbol{s}\|\leq\Delta_k}}{\text{minimize}}\, m_k(s) := \boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{H}_k\boldsymbol{s}, \tag{4.1.1}$$

   where $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{H}_k = \nabla^2 f(\boldsymbol{x}_k)$.

2. Compute

$$\rho_k = \frac{f(\boldsymbol{x}_k + \boldsymbol{s}_k) - f(\boldsymbol{x}_k)}{m_k(\boldsymbol{0}) - m_k(\boldsymbol{s}_k)}.$$

3. If $\rho_k \geq c$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ and $\Delta_{k+1} = \min\{\gamma_{\text{inc}}\,\Delta_k, \Delta_{\max}\}$.

4. Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and $\Delta_{k+1} = \gamma_{\text{dec}}\,\Delta_k$.

end

For simplicity, we focus here on exact solves of the trust-region subproblem, though we note that inexact solves using Krylov subspace techniques are among the most used techniques in practice [143, 140]. We will comment on the inexact setting in Section 4.2.3.

4.1.2 Sub-optimal complexity bounds for Newton trust region

The first complexity analysis of trust-region methods appears due to Gratton, Sartenaer and Toint [79] and focused on finding an approximate first-order stationarity point in the sense of Question 1.1.1. First-order complexity results for trust-region schemes were formalized as such by Cartis et al. [28]. Second-order complexity bounds were provided for trust-region methods by Cartis et al. [29], through an analysis that also applies to Algorithm 7. We describe below the main complexity result that can be obtained for this algorithm.

Theorem 4.1.1 *Suppose that f is $\mathcal{C}^{2,2}$ with a Lipschitz continuous Hessian. Then Algorithm 7 reaches an (ϵ, ϵ_H)-stationary point (satisfying (1.1.3)) in at most*

$$\mathcal{O}\left(\max\left\{\epsilon^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\right\}\right) \tag{4.1.2}$$

iterations.

The bound (4.1.2) was shown to be sharp for Algorithm 7, which may seem surprising as better bounds in $\mathcal{O}\left(\max\left\{\epsilon^{-2}, \epsilon_H^{-3}\right\}\right)$ can be obtained by gradient descent augmented with negative curvature directions [146, Section 9.3]. This improved bound can be derived for relatively minor modifications of Algorithm 7, such as choosing inexact steps at every iteration according to the criterion to improve [47], or busing two different trust regions to account for each criterion of (1.1.3) individually [76].

4.1.3 More on complexity of Newton-type methods

The optimal known complexity bound for a Newton-type method is $\mathcal{O}\left(\max\left\{\epsilon^{-3/2}, \epsilon_H^{-3}\right\}\right)$, which was first established for cubic regularization techniques [29, 121]. Similar guarantees were achieved by TRACE [51], a trust-region-based method, and by a hybrid variant combining trust-region and cubic regularization features [52].

Together with Stephen J. Wright, the author analyzed line-search variants of Newton's method, showing a bound in $\mathcal{O}\left(\max\left\{\epsilon^{-3}\epsilon_H^3, \epsilon_H^{-3}\right\}\right)$. Although not strictly equivalent to the one for cubic regularization, both bounds reduce to $\mathcal{O}(\epsilon^{-3/2})$ whenever the two terms in the max are set to be equal, i.e. when $\epsilon_H = \epsilon^{1/2}$. All methods of this form were later shown to belong to an optimal class of second-order schemes [32].

We note that the results for trust-region and cubic regularization schemes have been extended to the Riemannian setting, i.e. when the variable is constrained to lie in a Riemannian manifold [1, 19]. Several other complexity results have been proven in the Riemannian setting, with the difficulty lying in accounting for the Riemannian geometry of the problem in key assumptions such as Lipschitz continuity of the derivatives [18].

4.2 A trust-region Newton method with best known complexity results

This section is concerned with obtaining optimal complexity results for a trust-region variant as close as possible to the textbook variant [123, Chapter 4]. Together with Frank E. Curtis, Daniel P. Robinson, and Stephen J. Wright, the author proposed an algorithm that merely requires to add regularization to the trust-region subproblem in order to satisfy optimal complexity guarantees.

Section 4.2.1 describes the algorithm at hand, and its complexity is given in Section 4.2.2. Extensions of this algorithm, including inexact solves of the trust-region subproblem and related work, are discussed in Section 4.2.3.

4.2.1 An algorithm with regularized steps

Algorithm 8 differs from the standard method through the addition of a regularizing term in the subproblem objective [50]. This seemingly minor modification is key to guaranteeing complexity results, provided the regularization coefficient is set according to optimality tolerances.

Algorithm 8: Newton-trust region algorithm with regularization.

Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\Delta_0 > 0$, $\Delta_{\max} > \Delta_0$, $c \in (0, 1)$.
for $k = 0, 1, \ldots$ **do**

1. Compute a tentative step by solving the trust-region subproblem

$$\underset{\substack{\boldsymbol{s}\in\mathbb{R}^n \\ \|\boldsymbol{s}\|\leq\Delta_k}}{\text{minimize}}\, m_k(\boldsymbol{s}) + \frac{\epsilon_H}{2}\|\boldsymbol{s}\|^2, \tag{4.2.1}$$

   where $m_k(\boldsymbol{s})$ is defined as in (4.1.1).

2. Compute

$$\rho_k = \frac{f(\boldsymbol{x}_k + \boldsymbol{s}_k) - f(\boldsymbol{x}_k)}{m_k(\boldsymbol{0}) - m_k(\boldsymbol{s}_k)}.$$

3. If $\rho_k \geq c$, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$ and $\Delta_{k+1} = \min\{\gamma_{\text{inc}}\,\Delta_k, \Delta_{\max}\}$.

4. Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and $\Delta_{k+1} = \gamma_{\text{dec}}\,\Delta_k$.

end

4.2.2 Complexity results

At each iteration, Algorithm 8 minimizes a regularized version of problem (4.2.1). In particular, this certifies that any nonzero step will produce a model decrease in $\mathcal{O}(\epsilon_H\|\boldsymbol{s}\|^2)$, which is instrumental to deriving optimal iteration complexity bounds.

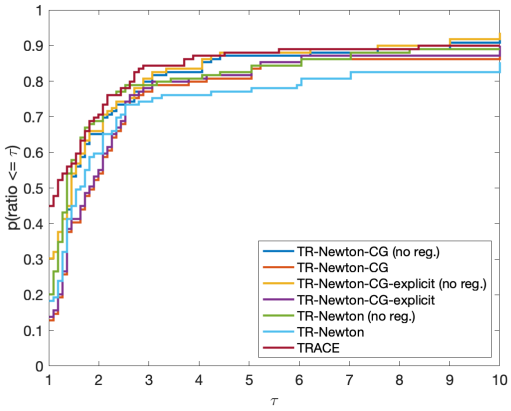
Theorem 4.2.1 *Suppose that f is $\mathcal{C}^{2,2}$ with a Lipschitz continuous Hessian. Then Algorithm 7 reaches an (ϵ, ϵ_H)-stationary point in at most*

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon^{-2}\epsilon_H, \epsilon_H^{-3}\right\}\right) \tag{4.2.2}$$

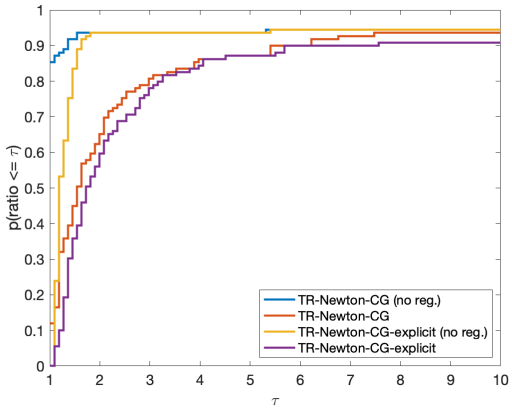
iterations.

When the two terms in the maximum of (4.2.2) are set to be equal, the bound (4.2.2) matches that of cubic regularization and other optimal Newton-type techniques [32].

Algorithm 8 can also been analyzed in an inexact fashion, by applying a Krylov-type technique to compute an approximate solution of (4.2.1). One natural candidate for this purpose is the truncated conjugate gradient method [140, 143], that applies conjugate gradient while stopping as soon as the trust-region boundary is reached. By combining this idea with the analysis of Section 3.2.2, one can design an inexact method with best-known complexity guarantees [50, Theorem 4.7]. A key to deriving these results is the use of a stronger regularization in the inexact variant than in the exact variant. More precisely, we replace the objective of (4.2.1) by $m_k(\boldsymbol{s}) + \epsilon_H\|\boldsymbol{s}\|^2$, i.e. we double the regularization parameter.



(a) Iterations for exact and inexact variants.



(b) Hessian-vector products of inexact variants.

Figure 4.1. Comparison of TRACE [51], standard Newton(-CG) trust-region schemes [41] and propsed Newton(-CG) trust-region schemes with regularization and complexity guarantees [50]. Experiments were conducted on a subset of CUTEst problems using $\epsilon = \sqrt{\epsilon_H} = 10^{-5}$.

4.2.3 The numerical impact of regularization

Our analysis requires regularization of order ϵ_H, which is a fair amount of regularization for standard tolerance values. Figure 4.1 presents experiments ran using $\epsilon = 10^{-5}$ and $\epsilon_H = \epsilon^{1/2} \approx 3.10^{-3}$ (recall that this represents the most favorable setting for our complexity bound (4.1.2)). The iteration profiles (leftmost figure) show that our approach with regularization does compare favorably with standard, non-regularized schemes. On the other hand, our inexact variants make a heavier use of Hessian-vector products (rightmost figure), due to the cost of applying capped CG (and, in particular, the strong convergence criterion). In a nutshell, these experiments illustrate the price paid for equipping a trust-region method with (optimal) complexity guarantees.

Overall, these experiments

4.3 Trust-region Newton methods for strict saddle problems

For general nonconvex $\mathcal{C}^{2,2}$ functions, the bounds seen in Sections 4.1 and 4.2 cannot be improved [32]. Nevertheless, on certain classes of nonconvex problems, gradient-based techniques can exhibit better complexity bounds, that are logarithmic in the optimality tolerances, akin to the bounds obtained for strongly convex optimization. Such properties include the Polyak-Łojasiewicz condition [149, Section 3.8], and typically involve first-order derivative information. Similar properties using second-order derivatives were proposed in the literature [48, 49], while scattered results appeared for specific nonconvex problems [141].

This section is based on a joint work with Florentin Goyens [70], in which we studied the complexity of (Riemannian) Newton trust-region methods when applied to a particular class of nonconvex problems called strict saddle problems. Section 4.3.1 defines this problem class, while Section 4.3.2 provides a complexity bound for the basic trust-region method of Section 4.1.1 that leverages the strict saddle structure. Other developments related to strict saddle functions are reviewed in Section 4.3.3.

4.3.1 The strict saddle paradigm

Together with Florentin Goyens [70], the author formalized the notion of a strict saddle function in the context of Riemannian optimization, so as to encompass a variety of problems defined over Riemannian manifolds. More precisely, we considered problems of the form

$$\underset{\boldsymbol{x}\in\mathcal{M}}{\text{minimize}}\, f(\boldsymbol{x}),$$

where $\mathcal{M}$ is a Riemannian manifold embedded in $\mathbb{R}^n$, and f is twice continuously differentiable in a Riemannian sense. Roughly speaking, the notions of gradient (first-order derivative), Hessian (second-order derivative), and Lipschitz continuity can be extended from the Euclidean setting $\mathcal{M} = \mathbb{R}^n$ to the Riemannian setting [18]. Consequently, one can both extend classical Euclidean algorithms to the Riemannian setting, and adapt their complexity analysis to handle manifold constraints. For the sake of both consistency and simplicity, however, we will present the results in the Euclidean case.

As in the rest of this chapter, we are concerned with computing approximate second-order stationary points in the sense of (1.1.3).

Definition 4.3.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable and let $\gamma, \lambda, \mu, \delta$ be positive constants. The function f is* $(\gamma, \lambda, \mu, \delta)$-strict saddle *if $\mathbb{R}^n = \mathcal{R}_{\mathrm{g}} \cup \mathcal{R}_{\mathrm{h}} \cup \mathcal{R}_{\mathrm{l}}$, where*

$$\begin{aligned}
\mathcal{R}_{\mathrm{g}} &= \{\boldsymbol{x} \in \mathbb{R}^n : \ \|\nabla f(\boldsymbol{x})\| \geq \gamma\} \\
\mathcal{R}_{\mathrm{h}} &= \{\boldsymbol{x} \in \mathbb{R}^n : \lambda_{min}\left(\nabla^2 f(\boldsymbol{x})\right) \leq -\lambda\} \\
\mathcal{R}_{\mathrm{l}} &= \{\boldsymbol{x} \in \mathbb{R}^n : \ \textit{there exists } \boldsymbol{x}^* \in \mathbb{R}^n, \textit{ a local minimizer of } f \textit{ such that } \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \delta \textit{ and} \\
&\qquad f \textit{ is } \mu\textit{-strongly convex over the set } \{\boldsymbol{y} \in \mathbb{R}^n : \ \|\boldsymbol{x}^* - \boldsymbol{y}\| < 2\delta\}\}.
\end{aligned}$$

A strict saddle function thus possesses a particular *landscape*, that divides the space into three (possibly non-connected and/or overlapping) regions. Per the definition, any point with zero gradient must be either a strict saddle point if it belongs to $\mathcal{R}_{\mathrm{h}}$ (that is, a first-order stationary point that is not a second-order stationary point) or close to a local minimum if it belongs to $\mathcal{R}_{\mathrm{l}}$.

Examples of such strict saddle functions include strongly convex functions over $\mathbb{R}^n$, for which $\mathcal{R}_{\mathrm{h}} = \emptyset$. A simple example of strict saddle function is the one-dimensional dimension $\phi : x \mapsto (x^2 - 1)^2$. Note that this function is semialgebraic, and as such falls into other nice classes of nonconvex functions [5]. Rayleigh quotient minimization over the sphere and complex instances of the phase retrieval problem are other instances of strict saddle functions [70, 142].

4.3.2 Complexity results to the strict saddle setting

The purpose of our work with Florentin Goyens [70] was the study of a popular method (trust region) in the specific context of strict saddle problems. We will thus consider the following assumption.

Assumption 4.3.1

(i) The function f is twice continuously differentiable with Lipschitz continuous Hessian.

(ii) The function f is $(\gamma, \lambda, \mu, \delta)$-strict saddle.

We emphasize that the second part of Assumption 4.3.1 is not required for the optimization process to be well-defined. In fact, we now present a simplified version of our main complexity result [70, Theorem 3.18] that applies to Algorithm 7 as stated in Section 4.1.1.

Theorem 4.3.1 *Suppose that Assumption 4.3.1 holds. Then, Algorithm 7 reaches an (ϵ, ϵ_H)-stationary point in at most $N_f + N_\epsilon$ evaluations, where*

$$\begin{cases} N_f &= \mathcal{O}\left(\max\{1, \gamma^{-2}, \gamma^{-4/3}\lambda^{-1}, \gamma^{4/3}\mu^{-1}, \gamma^{-2/3}\mu^{-2}, \lambda^{-3}, \lambda^{-2}\mu^{-1}, \lambda^{-1}\mu^{-2}, \mu^{-3}, \mu^{-2}\delta^{-1}\right) \\ N_\epsilon &= \mathcal{O}\left(\log\log\left(\epsilon^{-1}\right)\right). \end{cases} \tag{4.3.1}$$

iterations.

The bound (4.3.1) decomposes into two parts. The first term does not depend on optimality tolerances ϵ, ϵ_H, but rather on problem specific constants. This result is reminiscent of guarantees for Newton-type algorithms in the strongly convex setting [20, Section 9.6]. The second term is doubly logarithmic in ϵ^{-1}, which is a significant improvement over the result of both Theorem 4.1.1 and Theorem 4.2.1, for which the dependency is polynomial in ϵ^{-1}.

4.3.3 Further work on landscape-aware algorithms

Definition 4.3.1 originates from Ge et al. [65], that was concerned with the particular problem of tensor decomposition. A number of nonconvex instances have since then been analyzed in a similar fashion, in order to distinguish them from arbitrary nonconvex instances. Applications involving matrix variables, such as matrix sensing [14] and low-rank matrix completion [66]. Sun et al. [141] investigated a number of problems and showed that they satisfied a strict saddle property. Numerous examples have been collected and explained by Wright and Ma [146].

Provided all saddle points are strict, gradient-type methods such as gradient descet) can be certified to converge, sometimes with a good complexity, to second-order stationary point, thereby alleviating the need for tailored algorithms [105, 91]. Still, a number of landscape-aware algorithms have been proposed in the literature. The method of Paternain et al. [126] was designed for functions that satisfy a form of strict saddle property, in the sense that the Hessian cannot have eigenvalues arbitrarily small in magnitude. Sun et al. [142] studied a trust-region algorithm (with fixed trust-region radius) for a strict saddle formulation of phase retrieval. Though less focused on strict saddle, the general framework of Curtis and Robinson [49] as well as the phase complexity analysis of Cartis et al. [33, Section 5.4] fall into this category. Florentin Goyens and the author also proposed a landscape aware method using again a capped CG routine to solve the trust-region subproblem. Provided the strict saddle parameters are known, the algorithm chooses an appropriate step and possibly an appropriate regularization parameter according to those constants [70, Algorithm 4].

4.4 Conclusions and perspectives for Chapter 4

Newton-type methods are more challenging to analyze from a complexity viewpoint than gradient-based (or even derivative-free) techniques. Since those algorithms rely on second-order information, it appears natural to consider second-order stationary points as targets of those methods. In fact, this setting, that amounts to considering Question 1.1.2 in lieu of Question 1.1.1, appears to be the one where second-order methods provably possess better guarantees than standard first-order

algorithms. A key to deriving such results lies in a careful use of negative curvature directions. In particular, regularizing the objective appears to be a way to leverage significant negative curvature when present.

The algorithm proposed with Frank E. Curtis, Daniel P. Robinson and Stephen J. Wright [50] achieves optimal complexity results thanks to regularization. However, such regularization may increase the method's practical cost. Building a hybrid method allowing for the computation of unregularized steps (and, in particular, actual Newton steps when possible) could be a way to make this method even closer to the classical ones. Competitors such as the conjugate residuals-trust region method [53] could also be investigated from a complexity perspective.

Florentin Goyens' postdoctoral work with the author [70] established that classical Newton trust-region methods could be endowed with better complexity bounds when applied to strict saddle nonconvex functions. Although the properties of strict saddle functions are particularly suitable in optimization algorithms, certain nonconvex formulations such as overparameterized models do not satisfy our definition. Several proposals have looked into other nonconvex landscapes for which fast local convergence rates could be established [133], and this research direction is worth pursuing in order to tackle functions whose symmetries introduce non-isolated minima.

Chapter 5

Conclusion: From complexity to structures

In this final chapter, we briefly reflect on the contents of this manuscript, and how they highlight the author's research agenda and supervision capabilities. We then highlight several middle-term to long-term research perspectives that the author plans to consider for the next stage of his career.

5.1 Summary of the manuscript

This manuscript highlighted contributions of its author posterior to the PhD, that revolved around complexity of nonconvex optimization algorithms. Chapter 2 deals with nonconvex derivative-free optimization, the author's first research topic that continues to be an interest of his. As a member of the derivative-free optimization (DFO) community, the author has been part of a growing line of research on subspace methods, as described in Section 2.2. In parallel, the author co-supervised the PhD thesis of Sébastien Kerleau (defence scheduled in Fall 2025), that lead to precise evaluations of dimension dependences in complexity bounds.

Chapter 3 focuses on gradient-based methods, a topic tackled by the author through linear and nonlinear conjugate gradient techniques. The former allowed to design new eigenvalue approximation techniques for indefinite matrices, a key ingredient for generic nonconvex problems. These results, developed during the author's postdoctoral work, have received significant attention from the nonconvex optimization community. The author has maintained interest in that space, with the goal of developing algorithms as close as possible to textbook efficient nonlinear optimization techniques but with optimal complexity guarantees. Rémi Chan--Renous-Legoubin's master internship produced such a method, of nonlinear conjugate gradient type.

Chapter 4 is concerned with Newton-type methods, and in particular those with second-order guarantees. The author's interest for such algorithms grew out of part of his PhD work on second-order derivative-free algorithms [75, 76]. However, his primary contributions in the filed occurred during his postdoctoral years, and culminated in a joint publication between his posdoctoral advisor and two external faculty researchers [50]. Having tackled generic nonconvex problems, the author turned to specific problems structures while supervising Florentin Goyens (postdoctoral researcher in the author's group from 2022 to 2024). By leveraging Florentin Goyens's expertise in Riemannian optimization, we were able to define a broad class of problems of interest and to show improved guarantees for classical schemes on these problems, thanks to their favorable landscape structure.

5.1.1 Perspectives: Leveraging structures

A significant portion of the author's research has focused on generic, abstract nonconvex formulations, in which one can only rely on oracle information (function and derivative values) to perform optimization. Although this is a valuable paradigm for developing general purpose techniques, the ability to exploit special structure is key to obtain efficient methods in practice. It is even more important when one is concerned with a specific application or a problem, since that problem's specifics can be leveraged for efficient solves.

In this section, the author identifies several broad perspectives for his research, based on current departures from his existing works and longer-term perspectives.

5.1.2 Perspective I: Structure for nonconvex problems

The postdoctoral work of Florentin Goyens [70] focused on manifold optimization and specific (strict saddle) structures. Complexity results can reflect this favorable structure when it exists, as is the case on eigenvalue problems, for instance (recall the results of Section 3.2).

Meanwhile, identifying the gap between well-structured nonconvex instances and badly structured instances remains a challenge. The PhD thesis of Iskander Legheraba (defence scheduled in September 2025) highlighted challenges in these approaches. In this thesis, the seemingly simple problem of matrix square root approximation

$$\underset{\boldsymbol{X}\in\mathbb{R}^{n\times n}}{\text{minimize}}\ \frac{1}{2}\left\|\boldsymbol{X}^2-\boldsymbol{M}\right\|_F^2, \tag{5.1.1}$$

where $\boldsymbol{M}\in\mathbb{R}^{n\times n}$ is an arbitrary data matrix, was considered. Despite existing algorithmic proposals dedicated to solving this problem when $\boldsymbol{M}\succ\boldsymbol{0}$, there are no available proof for the strict saddle nature of problem (5.1.1). Moreover, there exist matrices $\boldsymbol{M}$ such that the problem is provably not strict saddle [107].

To encourage the development of algorithms dedicated to strict saddle problems, akin to the rise of convex optimization techniques, precise examples and benchmarks seem necessary. A classification effort of nonconvex problems was already suggested by the optimization community [141, 49], yet the lack of unified terminology and examples is an obstacle to developing a research community on this topic. To move this concept one step further, the author believes that a good subclass of nonconvex optimization problems should feature both toy examples and established benchmarks. To this end, the author submitted an ANR proposal (whose results are still pending at the time of this manuscript), that he hopes will allow him to push this topic further.

5.1.3 Perspective II: Discrete structures

The PhD thesis of Sébastien Kerleau revealed important connections between positive spanning sets and strongly connected digraphs, that have recently been published [46]. Further connections between the discrete mathematics community and the derivative-optimization one are likely to yield new optimization methods. The concept of positive k-spanning sets is one example among many. More broadly, we envision that polyhedral and geometrical techniques could benefit derivative-free integer programming, a field that remains relatively underexplored by the derivative-free optimization community.

Discrete structures have also led to advances in complexity. Submodular functions, originally a discrete notion for functions defined on finite sets, have been recently revisited from a continuous

perspective [8]. Interestingly, continuous submodular functions can be nonconvex, while possessing additional structure that allows for efficient optimization in a complexity sense [15]. Following a master internship supervision in 2024 (whose results will be presented at the conference ICCOPT 2025), the author plans to develop this research direction, both from the fundamental and the application side. Indeed, submodular optimization has found renewed interest from the machine learning community, and natural language processing in particular [16]. The author plans to build on the expertise in his department around that topic.

5.1.4 Perspective III: Structure beyond nonconvex problems

As stated in the introduction of this manuscript, complexity analysis in optimization arguably began with linear programmming. After established software and algorithms were built on interior-point methods, partly because of their favorable polynomial complexity guarantees [147], recent years have experienced a surge of interest in primal-dual first-order techniques. Those methods can be quite successful in extreme-scale problems due to their low per-iteration cost, while being amenable to a rich complexity analysis [3, 4]. A PhD thesis will be offered in October 2025 (with Antonin Chambolle as supervisor) on these aspects, with the goal of exploiting the specific structure of linear programs arising in optimal transport, a field that has attracted increasing attention in recent years [127].

Solving continuous linear programs is a typical building block for solving *integer* linear programs, typically through branch-and-bound techniques [104]. The author has been involved in an effort to solve a particular integer linear program that models office allocation space during building renovation at Université Paris Dauphine-PSL [2]. He was then able to assess the challenges posed by such programs, as well as the immediate benefits of leveraging structure. A research-oriented approach to this problem is currently under study in order to assess whether continuous algorithms (with good complexity properties) are suited as subroutines for solving this problem.

5.2 Final word: Research structures

After his PhD, the author was fortunate to conduct research in the Wisconsin Institute for Discovery (WID) as a postdoctoral researcher and then to LAMSADE as a faculty. In WID, he had numerous interactions with machine learning researchers during the rise of nonconvex optimization algorithms and landscape results in that community (starting 2016). The problems encountered by the community led to his focusing on nonconvex optimization during his postdoctoral studies [139, 138], then to his work on optimization landscapes and specific optimization problems [38, 70]. Those discussions continued within the machine learning team in LAMSADE, leading to the co-supervision of two PhD theses (Iskander Legheraba - 2020-2025, and Bastien Cavarretta since 2024) together with other team members.

While in WID, the author also had the opportunity to exchange with researchers in discrete mathematics, a topic of primary interest at LAMSADE. Having taken part in several events in WID, the author naturally started discussions with members of LAMSADE upon being recruited. Those ecled to the PhD thesis of Sébastien Kerleau (2021-2025), as well as a ongoing collaboration with colleagues around integer linear programming models [2].

In the future, the author plans to foster these interactions, as he believes they benefit to all parties involved while giving rise to both interesting problems and valuable applications. Building research networks will be at the core of his research agenda.

Bibliography

[1] A. Agarwal, N. Boumal, B. Bullins, and C. Cartis. Adaptive regularization with cubics on manifolds. *Math. Program.*, 188:85–134, 2021.

[2] S. Airiau, L. Galand, J. Lang, C. W. Royer, and S. Toubaline. Un modèle pour la cinématique du projet nouveau campus à Dauphine. In *ROADEF*, 2024.

[3] D. Applegate, M. Díaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. In *Advances in Neural Information Processing Systems*, volume 34, pages 20243–20257, 2021.

[4] D. Applegate, O. Hinder, H. Lu, and M. Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Math. Program.*, 201:133–184, 2023.

[5] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.*, 137:91–129, 2013.

[6] C. Audet, K. J. Dzahini, M. Kokkolaras, and S. Le Digabel. Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates. *Comput. Optim. Appl.*, 79:1–34, 2021.

[7] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, 2017.

[8] F. Bach. Submodular functions: from discrete to continuous domains. *Math. Program.*, 175:419–459, 2019.

[9] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization. *Math. Program.*, 134:223–257, 2012.

[10] A. S. Bandeira, K. Scheinberg, and L. N. Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.*, 24:1238–1264, 2014.

[11] A. S. Berahas, L. Cao, and K. Scheinberg. Global convergence rate analysis of a generic line search algorithm with noise. *SIAM J. Optim.*, 31:1489–1518, 2021.

[12] A. S. Berahas, M. J. O'Neill, and C. W. Royer. A line search framework with restarting for noisy optimization problems. arXiv:2506.03358, 2025.

[13] E. Bergou, E. Gorbunov, and P. Richtárik. Stochastic three points method for unconstrained smooth minimization. *SIAM J. Optim.*, 30:2726–2749, 2020.

[14] S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3880–3888, 2016.

[15] A. A. Bian, B. Mirzasoleiman, J. M. Buhmann, and A. Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Volume 54: Artificial Intelligence and Statistics, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 111–120. PMLR, 2017.

[16] J. A. Bilmes. Submodularity in machine learning and artificial intelligence. arXiv:2202.00132v1, 2022.

[17] J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence rate analysis of a stochastic trust region method via supermartingales. *INFORMS Journal on Optimization*, 1:92–119, 2019.

[18] N. Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, Cambridge, United Kingdom, 2023.

[19] N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.*, 39:1–33, 2019.

[20] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, United Kingdom, 2004.

[21] A. Brilli, M. Kimiaei, G. Liuzzi, and S. Lucidi. Worst case complexity bounds for linesearch-type derivative-free algorithms. *J. Optim. Theory Appl.*, 203:419–454, 2024.

[22] Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized nonconvex quadratic problems. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10726–10736, 2018.

[23] Y. Carmon and J. C. Duchi. Gradient descent finds the cubic-regularized non-convex Newton step. *SIAM J. Optim.*, 29:2146–2178, 2019.

[24] Y. Carmon and J. C. Duchi. First-order methods for nonconvex quadratic minimization. *SIAM Rev.*, 62:395–436, 2020.

[25] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. "Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the International Conference on Machine Learning, August 2017, Sydney, Australia*, pages 654–663, 2017.

[26] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points I. *Math. Program.*, 184:71–120, 2020.

[27] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Lower bounds for finding stationary points II: First-order methods. *Math. Program.*, 185:315–355, 2021.

[28] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM J. Optim.*, 20:2833–2852, 2010.

[29] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Complexity bounds for second-order optimality in unconstrained optimization. *J. Complexity*, 28:93–108, 2012.

[30] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM J. Optim.*, 22:66–86, 2012.

[31] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM J. Optim.*, 23(3):1553–1574, 2013.

[32] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, volume 3, pages 3697–3738, 2019.

[33] C. Cartis, N. I. M. Gould, and Ph. L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization: Theory, Computation and Perspectives*, volume MO30 of *MOS-SIAM Series on Optimization*. SIAM, 2022.

[34] C. Cartis and L. Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. *Math. Program.*, 199:461–524, 2023.

[35] C. Cartis and L. Roberts. Randomized subspace derivative-free optimization with quadratic models and second-order convergence. arXiv:2412.14431, 2024.

[36] C. Cartis, Ph. R. Sampaio, and Ph. L. Toint. Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization. *Optimization*, 64:1349–1361, 2015.

[37] C. Cartis and K. Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Program.*, 169:337–375, 2018.

[38] R. Chan–Renous-Legoubin and C. W. Royer. A nonlinear conjugate gradient method with complexity guarantees and its application to nonconvex regression. *Euro. J. Comput. Optim.*, 10:100044, 2022.

[39] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Math. Program.*, 169:447–487, 2018.

[40] Y. Chen, W. Hare, and A. Wiebe. Q-fully quadratic modeling and its application in a random subspace derivative-free method. *Comput. Optim. Appl.*, 89:317–360, 2024.

[41] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[42] A. R. Conn, K. Scheinberg, and L. N. Vicente. Geometry of interpolation sets in derivative-free optimization. *Math. Program.*, 111:141–172, 2008.

[43] A. R. Conn, K. Scheinberg, and L. N. Vicente. Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation. *IMA J. Numer. Anal.*, 28:721–748, 2008.

[44] A. R. Conn, K. Scheinberg, and L. N. Vicente. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM J. Optim.*, 20:387–415, 2009.

[45] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[46] D. Cornaz, S. Kerleau, and C. W. Royer. A characterization of positive spanning sets with ties to strongly connected digraphs. *Discrete Appl. Math.*, 374:105–119, 2025.

[47] F. E. Curtis, Z. Lubberts, and D. P. Robinson. Concise complexity analyses for trust region methods. *Optim. Lett.*, 12:1713–1724, 2018.

[48] F. E. Curtis and D. P. Robinson. How to characterize the worst-case performance of algorithms for nonconvex optimization. Technical Report 18T-003, COR@L Laboratory, Department of ISE, Lehigh University, 2018.

[49] F. E. Curtis and D. P. Robinson. Regional complexity analysis of algorithms for nonconvex optimization. *Math. Program.*, 187:579–615, 2021.

[50] F. E. Curtis, D. P. Robinson, C. W. Royer, and S. J. Wright. Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization. *SIAM J. Optim.*, 31:518–544, 2021.

[51] F. E. Curtis, D. P. Robinson, and M. Samadi. A trust region algorithm with a worst-case iteration complexity of $O\left(\epsilon^{-3/2}\right)$ for nonconvex optimization. *Math. Program.*, 162:1–32, 2017.

[52] F. E. Curtis, D. P. Robinson, and M. Samadi. An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *IMA J. Numer. Anal.*, 39:1296–1327, 2019.

[53] M.-A. Dahito and D. Orban. The conjugate residual method in linesearch and trust-region methods. *SIAM J. Optim.*, 29:1988–2025, 2019.

[54] Y.-H. Dai. Conjugate gradient methods with armijo-type line searches. *Acta Mathematicae Applicatae*, 18:123–130, 2002.

[55] A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *Foundations and Trends in Optimization*, 5:1–245, 2021.

[56] C. Davis. Theory of positive linear dependence. *Amer. J. Math.*, 76:733–746, 1954.

[57] M. Dodangeh, L. N. Vicente, and Z. Zhang. On the optimal order of worst case complexity of direct search. *Optim. Lett.*, 10:699–708, 2016.

[58] S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, B. Póczos, and A. Singh. Gradient descent can take exponential time to escape saddle points. In *Advances in Neural Information Processing Systems*, 2017.

[59] K. J. Dzahini. Expected complexity analysis of stochastic direct-search. *Comput. Optim. Appl.*, 81:179–200, 2022.

[60] K. J. Dzahini, M. Kokkolaras, and S. Le Digabel. Constrained stochastic blackbox optimization using a progressive barrier and probabilistic estimates. *Math. Program.*, 198:675–732, 2023.

[61] K. J. Dzahini, F. Rinaldi, C. W. Royer, and D. Zeffiro. Revisiting theoretical guarantees of direct-search methods. arXiv:2403.05322v2, 2024.

[62] K. J. Dzahini and S. M. Wild. Stochastic trust-region algorithm in random subspaces with convergence and expected complexity analyses. *SIAM J. Optim.*, 34:2671–2699, 2022.

[63] K. J. Dzahini and S. M. Wild. Direct search for stochastic optimization in random subspaces with zeroth-, first-, and second-order convergence and expected complexity. arXiv:2403.13320, 2024.

[64] R. Garmanjani, D. Júdice, and L. N. Vicente. Trust-region methods without using derivatives: Worst case complexity and the non-smooth case. *SIAM J. Optim.*, 26:1987–2011, 2016.

[65] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points - Online stochastic gradient for tensor decomposition. In *Volume 40: Conference on Learning Theory, 3-6 July 2015, Paris, France*, pages 797–842. PMLR, 2015.

[66] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minima. In *Advances in Neural Information Processing Systems 29*, pages 2973–2981, 2016.

[67] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.*, 2:21–42, 1992.

[68] G. H. Golub and C. F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, Baltimore, fourth edition, 2013.

[69] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads. *Comput. Optim. Appl.*, 60:545–557, 2015.

[70] F. Goyens and C. W. Royer. Riemannian trust-region methods for strict saddle functions with complexity guarantees. *Math. Program.*, 2024.

[71] G. N. Grapiglia. Quadratic regularization methods with finite-difference gradient approximations. *Comput. Optim. Appl.*, 85:683–703, 2023.

[72] G. N. Grapiglia. Worst-case evaluation complexity of a derivative-free quadratic regularization method. *Optim. Lett.*, 18:195–213, 2024.

[73] S. Gratton, S. Jerad, and Ph. L. Toint. Complexity of a class of first-order objective-function-free optimization algorithms. *Optim. Methods Softw.*, pages 1–31, 2024.

[74] S. Gratton, S. Jerad, and Ph. L. Toint. A stochastic objective-function-free adaptive regularization method with optimal complexity. *Open Journal on Mathematical Optimization*, 6:1–24, 2025.

[75] S. Gratton, C. W. Royer, and L. N. Vicente. A second-order globally convergent direct-search method and its worst-case complexity. *Optimization*, 65:1105–1128, 2016.

[76] S. Gratton, C. W. Royer, and L. N. Vicente. A decoupled first/second-order steps technique for nonconvex nonlinear unconstrained optimization with improved complexity bounds. *Math. Program.*, 179:195–222, 2020.

[77] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic descent. *SIAM J. Optim.*, 25:1515–1541, 2015.

[78] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. Direct search based on probabilistic feasible descent for bound and linearly constrained problems. *Comput. Optim. Appl.*, 72:525–559, 2019.

[79] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.

[80] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.*, 16:170–192, 2005.

[81] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.*, 2:35–58, 2006.

[82] W. Hare and G. Jarry-Bolduc. A deterministic algorithm to compute the cosine measure of a finite positive spanning set. *Optim. Lett.*, 14:1305–1316, 2020.

[83] W. Hare, G. Jarry-Bolduc, S. Kerleau, and C. W. Royer. Using orthogonally structured positive bases for constructing positive k-spanning sets with cosine measure guarantees. *Linear Algebra Appl.*, 680:183–207, 2024.

[84] W. Hare, G. Jarry-Bolduc, and C. Planiden. Nicely structured positive bases with maximal cosine measure. *Optim. Lett.*, 17:1495–1515, 2023.

[85] C. He, H. Huang, and Z. Lu. A Newton-CG based barrier-augmented lagrangian method for general nonconvex conic optimization. *Comput. Optim. Appl.*, 89:843–894, 2024.

[86] C. He, H. Huang, and Z. Lu. Newton-CG methods for nonconvex unconstrained optimization with hölder continuous hessian. *Math. Oper. Res.*, 2025.

[87] C. He and Z. Lu. A Newton-CG based barrier method for finding a second-order stationary point of nonconvex conic optimization with complexity guarantees. *SIAM J. Optim.*, 33:1191–1222, 2023.

[88] C. He, Z. Lu, and T. K. Pong. A Newton-CG based augmented lagrangian method for finding a second-order stationary point of nonconvex equality constrained optimization with complexity guarantees. *SIAM J. Optim.*, 33:1734–1766, 2023.

[89] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[90] M. Hough and L. Roberts. Model-based derivative-free methods for convex-constrained optimization. *SIAM J. Optim.*, 32:2552–2579, 2022.

[91] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. In *Volume 70: International Conference on Machine Learning, 6-11 August 2017, International Convention Centre, Sydney, Australia*, pages 1724–1732. PMLR, 2017.

[92] C. Jin, P. Netrapalli, and M. I. Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Proceedings of the 31st Conference On Learning Theory*, pages 1042–1085. PMLR, 2018.

[93] C. Jones and M. McPharlon. Spherical discrepancy minimization and algorithmic lower bounds for covering the sphere. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 874–891, 2020.

[94] S. Karimi and S. A. Vavasis. A unified convergence bound for conjugate gradient and accelerated gradient. arXiv:1605.00320, 2016.

[95] S. Karimi and S. A. Vavasis. Detecting and correcting the loss of independence in nonlinear conjugate gradient. arXiv:1202.1479v2, 2018.

[96] C. T. Kelley. *Implicit Filtering*. Software Environment and Tools. Society for Industrial and Applied Mathematics, Philadelphia, 2011.

[97] S. Kerleau. *Graphical and geometrical perspectives on positive spanning sets with applications to derivative-free optimization*. PhD thesis, Université Paris Sciences et Lettres, 2025.

[98] T. G. Kolda. Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM J. Optim.*, 16:563–586, 2005.

[99] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[100] J. Konečný and P. Richtárik. Simplified complexity analysis of simplified direct search. Technical Report ERGO 14-012, School of Mathematics, Edinburgh University, 2014.

[101] D. Kozak, S. Becker, A. Doostan, and L. Tenorio. A stochastic subspace approach to gradient-free optimization in high dimensions. *Comput. Optim. Appl.*, 79:339–368, 2021.

[102] D. Kozak, C. Molinari, L. Rosasco, L. Tenorio, and S. Villa. Zeroth-order optimization with orthogonal random directions. *Math. Program.*, 199:1179–1219, 2023.

[103] J. Larson, M. Menickelly, and S. M. Wild. Derivative-free optimization methods. *Acta Numer.*, 28:287–404, 2019.

[104] J. Lee. *A First Course in Linear Optimization*. Reex Press, fourth edition, version 4.07 edition, 2013-22.

[105] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht. First-order methods almost always avoid strict saddle points. *Math. Program.*, 176:311–337, 2019.

[106] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Volume 49: Conference on Learning Theory, 23-26 June 2016, Columbia University, New York, New York, USA*, pages 1246–1257. PMLR, 2016.

[107] I. Legheraba. *Landscape and complexity analyses for classes of nonconvex optimization problems*. PhD thesis, Université Paris Sciences et Lettres, 2025.

[108] R. M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical Report 96-71, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, 1996.

[109] D. C. Liu and J. Nocedal. On the limited-memory BFGS method for large scale optimization. *Math. Program.*, 45:503–528, 1989.

[110] Y. Liu and F. Roosta. MINRES: From negative curvature detection to monotonicity properties. *SIAM J. Optim.*, 32:2636–2661, 2022.

[111] D. A. Marcus. Minimal positive 2-spanning sets of vectors. *Proceedings of the AMS*, 82:165–172, 1981.

[112] D. A. Marcus. Gale diagrams of convex polytopes and positive spanning sets of vectors. *Discrete Appl. Math.*, 9:47–67, 1984.

[113] M. Menickelly. Avoiding geometry improvement in derivative-free model-based methods via randomization. arXiv:2305.17336, 2023.

[114] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Math. Program.*, 39:117–129, 1987.

[115] G. Nædvdal. Positive bases with maximal cosine measure. *Optim. Lett.*, 13:1381–1388, 2019.

[116] A. S. Nemirovski and D. B. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons, New York, 1983.

[117] Yu. Nesterov. A method for solving convex optimization problems with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[118] Yu. Nesterov. *Introductory lectures on convex optimization*. Kluwer Academic Publishers, Dordrecht, 2004.

[119] Yu. Nesterov. Random gradient-free minimization of convex functions. Technical Report 2011/1, CORE, Université Catholique de Louvain, 2011.

[120] Yu. Nesterov. *Lectures on convex optimization*. Springer International Publishing, second edition, 2018.

[121] Yu. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108:177–205, 2006.

[122] Yu. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17:527–566, 2017.

[123] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Ser. Oper. Res. Financ. Eng. Springer-Verlag, New York, second edition, 2006.

[124] M. O'Neill and Stephen J. Wright. Behavior of accelerated gradient methods near critical points of nonconvex problems. *Math. Program.*, 176:403–427, 2019.

[125] M. O'Neill and Stephen J. Wright. A line-search descent algorithm for strict saddle functions with complexity guarantees. In *ICML Workshop on Beyond First Order Methods in ML Systems*, 2020.

[126] S. Paternain, A. Mokhtari, and A. Ribeiro. A Newton-based method for nonconvex optimization with fast evasion of saddle points. *SIAM J. Optim.*, 29:343–368, 2019.

[127] G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11:355–607, 2019.

[128] C. J. Price and I. D. Coope. Frames and grids in unconstrained and linearly constrained optimization: a nonsmooth approach. *SIAM J. Optim.*, 14:415–438, 2003.

[129] R. Pytlak. *Conjugate Gradient Algorithms in Nonconvex Optimization*, volume 89 of *Nonconvex Optimization and Its Applications*. Springer-Verlag, Berlin Heidelberg, 2009.

[130] T. M. Ragonneau and Z. Zhang. An optimal interpolation set for model-based derivative-free optimization methods. *Optim. Methods Softw.*, 39:898–910, 2024.

[131] M. Rando, C. Molinari, S. Villa, and L. Rosasco. Stochastic zeroth order descent with structured directions. *Comput. Optim. Appl.*, 89:691–727, 2024.

[132] J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad, 2018.

[133] Q. Rebjock and N. Boumal. Fast convergence of trust-regions for non-isolated minima via analysis of CG on indefinite matrices. *Math. Program.*, 2024.

[134] R. G. Regis. On the properties of positive spanning sets and positive bases. *Optim. Eng.*, 17:229–262, 2016.

[135] R. G. Regis. On the properties of the cosine measure and the uniform angle subspace. *Comput. Optim. Appl.*, 78:915–952, 2021.

[136] L. Roberts and C. W. Royer. Direct search based on probabilistic descent in reduced spaces. *SIAM J. Optim.*, 33:3057–3082, 2023.

[137] Z. Romanowicz. Geometric structure of positive bases in linear spaces. *Appl. Math. (Warsaw)*, 19:557–567, 1987.

[138] C. W. Royer, M. O'Neill, and S. J. Wright. A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Math. Program.*, 180:451–488, 2020.

[139] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM J. Optim.*, 28:1448–1477, 2018.

[140] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.

[141] J. Sun, Q. Qu, and J. Wright. When are nonconvex problems not scary? arXiv:1510.06096, 2015.

[142] J. Sun, Q. Qu, and J. Wright. A geometric analysis of phase retrieval. *Found. Comput. Math.*, 18:1131–1198, 2018.

[143] Ph. L. Toint. Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, editor, *Sparse Matrices and Their Uses*, pages 57–88. Academic Press, London, 1981.

[144] L. N. Vicente. Worst case complexity of direct search. *EURO J. Comput. Optim.*, 1:143–153, 2013.

[145] R. F. Woltzlaw. *Incidence graphs and unneighborly polytopes*. PhD thesis, Technischen Universität Berlin, 2009.

[146] J. Wright and Y. Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[147] S. J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[148] S. J. Wright. Optimization algorithms for data analysis. In A. C. Gilbert M. W. Mahoney, J. C. Duchi, editor, *The mathematics of data*, number 25 in IAS/Park City Mathematics Series. AMS, IAS/Park City Mathematics Institute, and Society for Industrial and Applied Mathematics, Princeton, 2018.

[149] S. J. Wright and B. Recht. *Optimization for Data Analysis*. Cambridge University Press, 2022.

[150] Y. Xie and S. J. Wright. Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints. *J. Sci. Comput.*, 86:38, 2021.

[151] Y. Xie and S. J. Wright. Complexity of a projected Newton-CG method for optimization with bounds. *Math. Program.*, 207:107–144, 2024.

[152] Y.-X. Yuan. Recent avances in trust region algorithms. *Math. Program.*, 151:249–281, 2015.

RÉSUMÉ

L’optimisation vise à prendre la meilleure décision parmi un ensemble de possibilités. Les problèmes d’optimisation se modélisent par des objets mathématiques, résolus ensuite numériquement au moyen d’algorithmes. Lorsque ces problèmes sont convexes, les algorithmes sont souvent comparés à travers leurs bornes de complexité, dont le calcul est étudié depuis plus de cinquante ans. Dans le cas non convexe, en revanche, les analyses de complexité n’ont réellement pris leur essor que durant les quinze dernières années, et de nombreuses questions de recherche demeurent encore peu explorées.

Ce manuscrit présente les travaux de l'auteur en tant que chercheur indépendant et encadrant en analyse de complexité pour l'optimisation non convexe, et s'articule selon trois axes de recherche principaux. On s'intéresse tout d'abord à l'optimisation sans dérivées, où la dépendance des bornes de complexité en la dimension du problème est cruciale. On étudie ensuite les algorithmes de gradient conjugué, pour lesquels on identifie des modifications à même de conduire à des garanties de complexité. Enfin, on se concentre sur les algorithmes de Newton avec régions de confiance, dont la complexité varie naturellement selon la classe de problèmes non convexe considérée. Le manuscrit se conclut par une réflexion sur l'évolution de la recherche en analyse de complexité sur le long terme, à travers l'étude de diverses structures.

ABSTRACT

Optimization is concerned with making the best decision out of a set of alternatives. Optimization problems are modeled using mathematical objects, then solved using numerical algorithms. When those problems are convex, a common practice consists in comparing algorithms in terms of complexity guarantees, thereby using results that have been developed for more than fifty years. For nonconvex problems, however, complexity results have only grown in importance over the past fifteen years, while numerous research questions remain underexplored ;

This manuscript reviews its author’s contributions in the field of complexity analysis in nonconvex optimization, both as an independent researcher and as a supervisor. The manuscript is organized around three main research directions. We first focus on derivative-free optimization, where the dependency on the dimension is a crucial aspect of complexity guarantees. We then investigate conjugate gradient techniques, and describe changes to classical algorithmic templates that lead to complexity results. Finally, we study Newton trust-region methods, illustrating how their complexity results evolve with different subclasses of nonconvex optimization problems. We end the manuscript with long-term perspectives on complexity analysis, centered around the idea of structures.