THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *04/11/2016* **par :**

Clément W. ROYER

Derivative-Free Optimization Methods based on Probabilistic and Deterministic Properties: Complexity Analysis and Numerical Relevance

Algorithmes d'Optimisation Sans Dérivées à Caractère Probabiliste ou Déterministe : Analyse de Complexité et Importance en Pratique

---

JURY

Serge GRATTON	INPT-ENSEEIHT
Luís NUNES VICENTE	Universidade de Coimbra
Samir ADLY	Université de Limoges
Amir BECK	Technion Israel Institute of Technology
Jean-Baptiste CAILLAU	Université Bourgogne France-Comté
Anne GAZAIX	AIRBUS - IRT A. de Saint-Exupéry
Jean-Baptiste HIRIART-URRUTY	Université Toulouse 3 Paul Sabatier

---

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Serge GRATTON (Directeur) et *Luís NUNES VICENTE (Co-directeur)*
Rapporteurs :
Samir ADLY et *Amir BECK*
Président du Jury :
Jean-Baptiste HIRIART-URRUTY

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Randomization has had a major impact on the latest developments in the field of numerical optimization, partly due to the outbreak of machine learning applications. In this increasingly popular context, classical nonlinear programming algorithms have indeed been outperformed by variants relying on randomness. The cost of these variants is usually lower than for the traditional schemes, however theoretical guarantees may not be straightforward to carry out from the deterministic to the randomized setting. Complexity analysis is a useful tool in the latter case, as it helps in providing estimates on the convergence speed of a given scheme, which implies some form of convergence. Such a technique has also gained attention from the deterministic optimization community thanks to recent findings in the nonconvex case, as it brings supplementary indicators on the behavior of an algorithm.

In this thesis, we investigate the practical enhancement of deterministic optimization algorithms through the introduction of random elements within those frameworks, as well as the numerical impact of their complexity results. We focus on direct-search methods, one of the main classes of derivative-free algorithms, yet our analysis applies to a wide range of derivative-free methods. We propose probabilistic variants on classical properties required to ensure convergence of the studied methods, then enlighten their practical efficiency induced by their lower consumption of function evaluations. First-order concerns form the basis of our analysis, which we apply to address unconstrained and linearly-constrained problems. The observed gains incite us to additionally take second-order considerations into account. Using complexity properties of derivative-free schemes, we develop several frameworks in which information of order two is exploited. Both a deterministic and a probabilistic analysis can be performed on these schemes. The latter is an opportunity to introduce supplementary probabilistic properties, together with their impact on numerical efficiency and robustness.

Keywords: *derivative-free optimization, probabilistic properties, complexity analysis, direct-search methods.*

Résumé

L'utilisation d'aspects aléatoires a contribué de façon majeure aux dernières avancées dans le domaine de l'optimisation numérique; cela est dû en partie à la recrudescence de problèmes issus de l'apprentissage automatique (*machine learning*). Dans un tel contexte, les algorithmes classiques d'optimisation non linéaire, reposant sur des principes déterministes, se révèlent en effet bien moins performants que des variantes incorporant de l'aléatoire. Le coût de ces dernières est souvent inférieur à celui de leurs équivalents déterministes ; en revanche, il peut s'avérer difficile de maintenir les propriétés théoriques d'un algorithme déterministe lorsque de l'aléatoire y est introduit. Effectuer une analyse de complexité d'une telle méthode est un procédé très répandu dans ce contexte. Cette technique permet d'estimer la vitesse de convergence du schéma considéré et par là même d'établir une forme de convergence de celui-ci. Les récents travaux sur ce sujet, en particulier pour des problèmes d'optimisation non convexes, ont également contribué au développement de ces aspects dans le cadre déterministe, ceux-ci apportant en effet un éclairage nouveau sur le comportement des algorithmes.

Dans cette thèse, on s'intéresse à l'amélioration pratique d'algorithmes d'optimisation sans dérivées à travers l'introduction d'aléatoire, ainsi qu'à l'impact numérique des analyses de complexité. L'étude se concentre essentiellement sur les méthodes de recherche directe, qui comptent parmi les principales catégories d'algorithmes sans dérivées ; cependant, l'analyse sous-jacente est applicable à un large éventail de ces classes de méthodes. On propose des variantes probabilistes des propriétés requises pour assurer la convergence des algorithmes étudiés, en mettant en avant le gain en efficacité induit par ces variantes : un tel gain s'explique principalement par leur coût très faible en évaluations de fonction. Le cadre de base de notre analyse est celui de méthodes convergentes au premier ordre, que nous appliquons à des problèmes sans ou avec contraintes linéaires. Les bonnes performances obtenues dans ce contexte nous incitent par la suite à prendre en compte des aspects d'ordre deux. À partir des propriétés de complexité des algorithmes sans dérivées, on développe de nouvelles méthodes qui exploitent de l'information du second ordre. L'analyse de ces procédures peut être réalisée sur un plan déterministe ou probabiliste: la deuxième solution nous permet d'étudier de nouveaux aspects aléatoires ainsi que leurs conséquences sur l'efficacité et la robustesse des algorithmes considérés.

Mots-clés: *optimisation sans dérivées, propriétés probabilistes, analyse de complexité, méthodes de recherche directe.*

Acknowledgments

First and foremost, I am very thankful to my two advisors, Serge Gratton and Luís Nunes Vicente. Serge gave me the opportunity to start a thesis quite early, believing in the quiet ENSEEIHT student I was back then. He provided me with a great working environment, and always ended up being there when I really needed it. I learnt a lot about research, teaching and academia thanks to him (sometimes the hard way!). Luís welcomed me in the amazing city of Coimbra during my master thesis, introducing me to the world of optimization and the fruitfulness of international collaboration. This was the start of a three-year journey where I got to experience his kindness as well as his rigor, through numerous friendly yet intense discussions. I hope that this thesis reflects the positive influence both of them had on me, and I certainly expect it to continue in the future. *Merci pour tout*, Serge. *Muito obrigado*, Luís.

I am grateful to Professors Samir Adly and Amir Beck for agreeing to review this thesis. Thank you for your reports: I am honoured that you considered this work worthy of a PhD. I would like to thank Jean-Baptiste Hiriart-Urruty for acting as the president of the jury and perfectly chairing the defence: it was an honour to have you in my commitee. I also thank Anne Gazaix for agreeing to be an examiner of my work. I am glad that you took interest in my work, and I hope you got more familiar with the topic.

I am indebted to Jean-Baptiste Caillau, both for being part of my jury and giving me my very first experience with research. It was in the Mathematical Institute of Burgundy, in Dijon, and it was a milestone in my (short) career. Since then, Jean-Baptiste has been very supportive in every occasion our paths crossed, were it in Toulouse or Dijon. *Merci beaucoup*, Jean-Baptiste.

The work in this thesis started in collaboration with Zaikun Zhang in Coimbra and afterwards in Toulouse. His help during my first steps in the optimization community, as well as his friendship and support, were instrumental to my accomplishments, and he truly deserved to be cited here, on a professional but most of all on a humane point of view. 谢谢 , Zaikun.

I would also like to thank Stefan Wild and Jeffrey Larson, who welcomed me at Argonne National Laboratory during the beginning of 2016. In addition to being a first research experience in the US, it was a change of air I needed before diving back into the last months of my PhD in Toulouse. Thanks to you both.

This thesis was mostly written in my office at ENSEEIHT, and I would like to acknowledge all the people with whom I have shared this office, in appearing order:

Mohamed Zenadi, Florent Lopez, Théo Mary (probably the one that had the hardest time in my presence!), Damien Goubinat, Charlie Vanaret and Luce Le Gorrec. For all the discussions, the laughter, and the bitterness, thanks, guys. Thanks also to El Houcine Bergou, Youssef Diouane, Jean-Baptiste Latre, Nacer Soualmi and Thibaud Vandamme for our exchanges, at CERFACS or elsewhere.

During those years, I shared my time between research in the APO team at IRIT, and teaching in the IMA department of ENSEEIHT. For the first part, I thank the entire team for their support, particularly Chiara Puglisi for her kindness and Patrick Amestoy for all the early morning encounters. Sylvie "SAM" Armengaud-Metche, Sylvie Eichen and Annabelle Sansus kindly helped me not to get (too) lost in administrative paperwork: they definitely deserve my thanks. On the teaching side, I am grateful to our department chair Joseph Gergaud for the opportunities I had, and to Muriel De Guibert for her help. My thanks also go to Alfredo Buttari, Ronan Guivarch, Daniel Ruiz, Ehouarn Simon and Xavier Vasseur who helped me in learning this job, and eventually liking it. I am finally indebted to Patrick Altibelli for our collaboration prior to his retirement: it was certainly a privilege for me.

This thesis has been inspired by many people: not only those who I already mentioned, but people that I have interacted with during, before and after my PhD. Of course, the list starts with my family: my parents, my sister, my grandmother, as well as the Toulouse branch of the tree (uncle, aunt, cousins, great-cousins, the list keeps growing). Thank you for all the support you gave me at all times.

After six years in Toulouse, trying to mention all the people I met there and not to forget anyone would clearly be too risky. Let thus Alex, Baptiste and Pedro be your representatives of these years, a role they fully assumed during my defence. By thanking them, I (a.k.a., Geb) thank all of you for every moment we shared, in our school or elsewhere, even though there could have been a lot more of those.

A special thanks to my partner in crime, Pierre Marighetto, who accompanied me from Bordeaux to Toulouse to North America (well, that will be for 2017). Thanks to him, I opened up, came to realize what I was good at, and had some of the craziest and funniest times of my life. Until we meet again, old friend!

I also thank the trios that go way back: Alexia, Alice and Clément (the Bordeaux belote crew), Guillaume, Hugo and Olivier (the high school clan), Emilie, Lucile and Sébastien (the JBJ team). Even with distance, we were always connected, there is no reason for that to change.

Last but not least, I am deeply grateful to the musicians that I have met during the last years, mostly members of the Trous Balourds and the Gimmick Five Syndicate. Thank you for every rehearsal, every set, every note that I got to play, especially when I did not feel like it. I still have a long way to go, but this part of the road could not have been done without you by my side. You kept me in tune and rhythm, you backed me up on solos, and you certainly know how to get busy in love.

To the jokes that still make me smile

Thesis soundtrack

Breakbot - *By Your Side*

Eddie Daniels - *Morning Thunder*

Miles Davis & Robert Glasper - *Everything's Beautiful*

Electro Deluxe - *Hopeful, Play, Live in Paris, Home*

Gimmick Five Syndicate - *Live in Toulouse (December 8, 2012)*

Robert Glasper Trio - *Covered*

Robert Glasper Experiment - *Black Radio, Black Radio 2, ArtScience, Tiny Desk Concert, Live with the Metropol Orkest 2014*

Bill Laurance - *Flint, Swift, Aftersun*

Hocus Pocus - *73 touches, Place 54, 16 pièces*

Marcus Miller - *Renaissance, Afrodeezia, Live with the Metropol Orkest 2013*

Panam Panic - *The Black Monk*

Snarky Puppy - *Tell Your Friends, Ground Up, We Like It Here, Family Dinner Vol. 1 and 2, Culcha Vulcha, Live at Stockholm Jazz Festival 2013, Live at Java Jazz 2014, Live sets at the Funky Biscuit 2015, Lives World Tour 2015*

Snarky Puppy and the Metropol Orkest - *Sylva, Live in Coutances 2015*

Chapter 1

Introduction

Numerical optimization has recently faced a change of paradigm due to the outbreak of algorithms based on randomness. One of the main reasons for such an evolution lies in the efficiency of these randomized schemes in solving problems arising in machine learning applications [19]. Indeed, in those problems, randomized first-order algorithms have proved more effective than traditional deterministic frameworks. The large dimension of the problems to be addressed in machine learning is one possible source of improvement through randomization. Indeed, in such a large-scale setting, it is very likely that attempting to use the whole available data (say, in a gradient step) is unnecessary as the information may present some redundancy. In these situations, randomization might be of assistance.

The main drawback of randomization is that it can jeopardize the deterministic convergence properties of the algorithm at hand. Still, theoretical analysis of those schemes can be performed, one way being the derivation of complexity results. Given some convergence criterion, the goal of such an analysis is to provide a worst-case estimate of the rate of decrease of this criterion, deterministically or in expectation, which can then result in convergence guarantees. Although such global rates are common in deterministic or randomized optimization methods applied to convex problems, results in the nonconvex setting have only arisen in the last decade, mainly due to the analysis of cubic regularization methods to that respect [23].

In the case of complex functions, a supplementary issue may prevent the application of classical first and second-order methods: the unavailability of the derivatives within the algorithmic process. In this setting, one may resort to derivative-free algorithms, which we describe in the next paragraph. For such methods, the study of their potential for randomization and the interest in performing their complexity analysis is then a natural question.

A brief description of derivative-free optimization

The general problem we consider in this thesis is the minimization of a numerical function f of n real parameters. When no constraints are enforced on the problem variables, the

commonly adopted formulation is the following:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1.1}$$

The function f, called the *objective function*, is assumed to be computable at every point in $\mathbb{R}^n$, where n is a positive integer.

In many situations, this minimization problem cannot be solved with derivative-based state-of-the art numerical optimization methods. Indeed, typical algorithmic schemes such as line-search [101] or trust-region methods [33] are based on the gradient vector, which is the representation of the first-order derivative in the variable space. Those methods may additionally require the computation of the Hessian matrix, associated with the second-order derivative. Therefore, whenever access to derivative information is not possible, such algorithms cannot be applied. This is problematic as the objective function can present itself in a complex form that does not allow to compute the first-order derivative. This is for instance the case whenever the expense of a gradient computation is extremely elevated: the cost of this derivative can forbid its exploitation in an algorithmic framework. Another possibility is that the gradient simply does not exist, which can occur when one attempts to optimize a function that has different expressions depending on the intervals in which the optimization parameters lie. Last but not least, a common situation in practice is the so-called *black-box* problems, for which the objective function is available as an oracle, typically of the form of a proprietary simulation code. In that setting, the user can only input the variable x and obtain the corresponding function value as an output, without any insight on the computation of this value. As a result, no explicit formula for the derivative is available, which prevents the user from accessing the corresponding values. When the code itself is only available as an executable file, techniques such as automatic differentiation cannot even be considered.

In order to tackle the above issues, one thus has to rely on optimization algorithms that only exploit the objective values. Such methods are qualified as *zeroth-order* or *derivative-free*, and remain very attractive among practitioners who have limited knowledge about the structure of their problem. Indeed, the aforementioned techniques often rely on elementary ideas (such as direct comparison of the function values to determine a better one), thus facilitating their immediate, albeit naive, implementation. In the last two decades, convergence and complexity results have been established for a wide range of derivative-free algorithms, thereby motivating further their popularity in practice through theoretical justifications. *Derivative-free optimization* (DFO) is now well established as a optimization category of its own [38].

Numerous derivative-free methods are not deterministic, in that their algorithmic process involves a certain form of randomness. The corresponding algorithms are sometimes called stochastic or randomized algorithms, and are typically used for global optimality purposes, i.e. when the goal of the user is to compute the global optimum of the objective [116]. The stochastic nature of the algorithms enables the exploration of the whole variable space, by allowing to escape the vicinity of a local optimum [108]. However, it may also prevent them from exhibiting theoretical guarantees other than

those based on density arguments. They can also require a significant budget of function evaluations to perform efficiently, which can preclude their use in certain contexts. This partly explains why some of them are not endowed with a convergence analysis, nor are they studied from a worst-case point of view.

For these reasons, our starting point when designing our methods will be deterministic schemes with local optimality purposes, for which the theoretical analysis has been well established. Two paradigms are essentially of use in deterministic DFO algorithms. The first one encompasses the class of *model-based methods*, which are often inspired by existing schemes relying on derivatives (trust-region, Levenberg-Marquardt, etc). When in a derivative-based context, those algorithms can make use of the first-order (and sometimes second-order) Taylor expansion(s) to build such models. Derivative-free methods, on the other hand, must construct approximations of the objective function based solely on sampled evaluations of that function. Polynomial approximation techniques have proven useful in providing models with a quality comparable to that of the Taylor models [35, 36]. However, ensuring a good quality of such models is a complicated task, that has repercussions on both the formulation and implementation of these algorithms (see [38, Chapter 6] for more details). Still, we will present results for a simplified subclass of derivative-free trust-region methods in Chapters 4 and 7.

In the thesis, we will concentrate on direct-search-type methods, which form the second main class of deterministic DFO algorithms. Those frameworks proceed by exploring the space along appropriately selected sets of directions, and move towards points that are able to reduce the current function value. The simplicity of this approach partially explains why those methods are among the oldest in numerical optimization, and remain a popular approach to overcome the lack of derivatives. We provide a more detailed analysis of direct search in Chapter 2.

Contribution of the thesis

This thesis proposes a theoretical and numerical study of derivative-free algorithms in which classical features are relaxed by requiring properties to be satisfied in probability or in the aim of providing improved complexity guarantees.

The first main contribution of the thesis is the proposal of an analytic tool that enables to study direct-search algorithms based on randomly generated directions. We describe both a martingale argument that helps in deriving convergence results and a new complexity proof technique leading to probabilistic worst-case complexity bounds, which can be adapted to several other algorithmic frameworks relying on random elements. In the case of direct-search methods, both analyses can also be extended from the unconstrained setting (which constitutes the basis of our results) to the bound-constrained and linearly-constrained cases.

The second contribution of this work is the study of second-order aspects in derivative-free algorithms, mostly driven by their complexity analysis. We provide a detailed study of second-order behavior in direct-search methods, thereby deriving the first worst-case complexity bounds related to second-order optimality in DFO. Such a investigation is

then completed by the introduction of a de-coupling technique, that is analyzed for two classes of zeroth-order schemes: improvement of existing complexity bounds is established, and practical implications of the new approaches are discussed.

The third contribution we identify builds on the previous ones by looking at second-order algorithms relying on probabilistic properties. The de-coupled framework reveals itself prone to randomization, and we propose an hybridization of probabilistic and deterministic guarantees that is shown to improve numerical efficiency while enabling to prove convergence. While the first results of the thesis can directly be applied in that setting, additional quality of the directions with respect to a second-order criterion requires further study. We thus provide a general context in which randomization has a positive impact in theory, as well as a comparison on the difficulty of satisfying several specific instances of such a property.

Structure of the thesis

This thesis is structured in two parts, that mainly differ by the level of smoothness we consider for the targeted objective function.

The first part of the thesis (Chapters 2 to 5) is concerned with the probabilistic analysis of derivative-free methods, mostly of direct-search type, with first-order convergence in mind. We review the core features of these directional schemes in Chapter 2, where we provide the necessary ingredients for a deterministic analysis. We then propose to replace the main property required to derive proper theoretical proofs by a probabilistic equivalent. This is the purpose of Chapter 3, in which we detail the convergence and complexity results that can be obtained, while providing numerical illustrations of the interest of our approach. Chapter 4 describes the application of our original complexity proof technique to a derivative-free trust-region method, and contains a short introduction to this class of algorithms. In Chapter 5, we discuss the extension of the direct-search scheme to bounds and linear constraints problems, by proposing probabilistic strategies that can offer significant improvement over a state-of-the-art implementation.

In the second part of the thesis (Chapters 6 to 8), we attempt to make use of the economy induced by our probabilistic approach to consider additional local information, of second-order type. To do so, we first propose in Chapter 6 a thorough study of deterministic properties of direct-search methods, which identifies the key aspects that allow for a second-order convergence analysis. A method that additionally presents convergence guarantees is proposed, and its practical qualities on tackling nonconvex problems are established. We then take a first step towards probabilistic variants of a second-order reasoning, by developing a dissociated treatment of first and second-order information: this is done in Chapter 7 using both a trust-region and a direct-search framework. Such an approach is shown to be beneficial in terms of worst-case behavior and numerical robustness, especially as probabilistic treatment of the first-order aspects without compromising the second-order ones becomes possible. We complement this study by proposing in Chapter 8 a general probabilistic second-order assumption that encompasses various deterministic properties, and we discuss its practical significance in

a derivative-free setting.

Chapter 9 concludes the main body of the thesis by summarizing its findings and highlighting several perspectives for future research, and is followed by two appendices. Appendix A describes the tools and results from probability theory that are applied throughout the thesis, while Appendix B provides a list of the test problems used in our numerical experiments.

Terminology and notations

We list below several conventions that will be adopted in the rest of the thesis.

A *probabilistic analysis* of an algorithm is defined as a theoretical study (typically related to the convergence) of this algorithm, performed thanks to arguments from probability theory. The associated method may or may not rely on random elements, yet a probabilistic analysis could be derived in both cases as long as appropriate distributions can be defined. More generally, the adjective *probabilistic* will qualify a concept or a result that involves probability theory.

A *randomized algorithm* is a method obtained from a deterministic one by introducing random aspects. In that sense, the algorithms to be developed in this thesis can be considered as randomized. Nevertheless, it can still be possible to derive a *deterministic analysis* of a randomized method. Since we are concerned with probabilistic analysis of the algorithms we propose, we will not refer to our algorithms as being randomized, but rather put the emphasis on the probabilistic properties that characterize them.

We qualify an optimization algorithm as *globally convergent* when its convergence properties do not depend on the chosen starting point, as opposed to locally convergent methods [101]. This property is related to convergence towards *local* first-order or second-order stationary points.

Notations Throughout the thesis, any set of vectors $V = \{v_1, \dots, v_m\} \in (\mathbb{R}^n)^m$ will be identified to the matrix $[v_1 \cdots v_m] \in \mathbb{R}^{n \times m}$. We will thus allow ourselves to write $v \in V$ even though we may manipulate V as a matrix. For a set of vectors V, we will denote by $\operatorname{span}(V)$ the subspace generated by linear combinations of the vectors in V, and by $\operatorname{pspan}(V)$ the set of nonnegative linear combinations of those vectors. We will use $\|\cdot\|$ to denote both the Euclidean norm of a vector $x \in \mathbb{R}^n$, and the associated matrix norm $\|A\| = \max_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|}$ for a matrix $A \in \mathbb{R}^{n \times n}$. For a symmetric matrix H, we will use $\lambda_{\min}(H)$ to denote its minimum eigenvalue.

When considering sequences arising from the execution of an optimization algorithm, we will use a subscript to indicate the related iteration. More generally, $\{i, j, k, l\}$ will be our preferred notations for indexes of (sub)sequences.

Finally, the notation $\mathcal{O}(Q)$ will indicate a constant times Q, where the constant contained in $\mathcal{O}(\cdot)$ does not depend explicitly on the quantities present in Q.

Chapter 2

Directional direct-search optimization methods

This thesis presents an analysis essentially revolving around the *direct-search* class of optimization methods. In addition to forming one of the most popular categories of derivative-free algorithms, it offers simple frameworks in which the main concepts of the probabilistic analysis derived in the thesis can be clearly enlightened.

In this chapter, we describe in detail a class of direct-search methods, with the associated theoretical results. We first provide useful tools to cope with the absence of derivatives and describe the main algorithmic framework in Section 2.1. Section 2.2 presents the associated convergence theory of the method, while Section 2.3 details the recent advances related to its complexity properties. Section 2.4 gives a brief overview of other methods belonging to the direct-search class.

2.1 General directional direct-search algorithmic framework

Direct-search methods are characterized by the fact that they rely solely on function value comparison to look for the optimum of an objective function. The set of directions to be used for selecting new points is one fundamental component of the method, so is the process that defines and iteratively updates the step size.

In this section, we describe the basic concept of a direct-search method of directional type. The presented algorithm is slightly less general than the one that can be found in classical references [37, 75], but we adopt this formulation in order to conform with the methods that will be developed throughout the thesis. Beforehand, we develop preliminary notions that are fundamental in the theoretical analysis of our class of methods.

2.1.1 Positive spanning sets and cosine measure

In direct-search methods, the variable space is explored through suitably chosen sets of directions. In order for those directions to yield sufficiently relevant information about the whole space, it is often required that they form a Positive Spanning Set, as defined below.

Definition 2.1 *Let $D = \{d_1, \ldots, d_m\} \subset \mathbb{R}^n$ a set of vectors in $\mathbb{R}^n$. D is called a* Positive Spanning Set (PSS) *if it can span the entire space by nonnegative linear combinations.*

Definition 2.2 *A set of vectors is called* positively independent *if no vector in the set can be expressed as a nonnegative linear combination of the others.*

A Positive Spanning Set D is called a positive basis *if it is a set of positively independent vectors.*

Example 2.1 *Let $\{e_1, \ldots, e_n\}$ be the canonical basis of $\mathbb{R}^n$. Then*

i) $D_{\oplus} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\}$ *is a positive basis.*

ii) $V_{n+1} = \left\{e_1, \ldots, e_n, -\frac{1}{\sqrt{n}} \sum_{i=1}^{n} e_i\right\}$ *is a positive basis.*

The earliest rigorous study of PSS was performed by Davis [43]. We mention below its essential features.

Proposition 2.1 *Let $D = \{d_1, \ldots, d_m\}$ be a set of vectors in $\mathbb{R}^n$.*

i) If D is a PSS, $m \geq n + 1$;

ii) If D is a positive basis, $n + 1 \leq m \leq 2\,n$.

Positive bases with $n + 1$ and $2\,n$ vectors are called *minimal* and *maximal* positive bases, respectively. One thus sees that there is a lower bound on the size of a PSS, which is of order of the problem dimension.

By considering a PSS (or a positive basis), one may thus resort only to nonnegative linear combinations of vectors in this set and still be able to generate the whole space. This already is an interesting property, but in the prospect of approximating specific vectors in the space, we would like to have a measure of the quality of a given set of directions (not necessarily a PSS). This is the sense of Definition 2.3.

Definition 2.3 *Let D be a set of (nonzero) vectors in $\mathbb{R}^n$. We define the* cosine measure *of D as the quantity*

$$\mathrm{cm}(D) \;=\; \min_{\substack{v\in\mathbb{R}^n\\ v\neq 0}} \max_{d\in D} \frac{d^\top v}{\|d\|\|v\|}, \tag{2.1}$$

which lies between -1 and 1.

The cosine measure quantifies the ability of a given set to approximate any vector in the associated space.

There is an important link between the cosine measure and the positive spanning set property, as established by the following proposition.

Proposition 2.2 *D is a Positive Spanning Set in $\mathbb{R}^n$ if and only if* $\mathrm{cm}(D) > 0$.

In other words, for any vector in the space and any PSS, there is always one element of the PSS that makes an acute angle with this vector. This is of particular interest since one of the goals of optimization schemes is to determine *descent directions*, that by definition make an acute angle with the negative gradient [101].

2.1.2 Algorithmic description

Algorithm 2.1 presents the basic scheme of our method, enlightening the process of one iteration. The method may start with a *Search Step*, in which several points are chosen and the poll step is skipped if one of those points yields an improved function value, as determined by (2.2). If none of those points happens to improve the objective, then a *Poll Step* is applied. Its decrease condition (2.3) is the same as for the search step, yet this step is fundamentally different. Indeed, the (polling) directions must possess specific properties in order to guarantee convergence of the framework: those are not required for the search directions.

Depending on whether a better point has been encountered or not, the iteration is identified as *successful* or *unsuccessful*. In the first case, the new point becomes the next iterate, and the step size is (possibly) increased; in the second one, the current point (also called the current iterate) does not change, and the step size has to be reduced.

We make several remarks about this method. First, the search step present in Algorithm 2.1 is optional in that it does not intervene in the theoretical analysis of the method. It can therefore be ignored for the convergence and complexity proofs. This being said, we point out that such a step is frequently used by practitioners, e.g. to test trial steps generated outside of the direct-search process, and that it may be the actual step of interest in some cases.

Algorithm 2.1: Basic directional Direct-Search method (BDS)

Choose an initial point $x_0 \in \mathbb{R}^n$, as well as $0 < \theta < 1 \leq \gamma$, $0 < \alpha_0 < \alpha_{\max}$.
Define a *forcing function* $\rho : \mathbb{R}^+ \to \mathbb{R}^+$.
for $k = 0, 1, 2, \ldots$ **do**
  Search Step (optional)
  Determine a set of search directions D_k^s in $\mathbb{R}^n$. If there exists $d \in D_k^s$ such that

$$f(x_k + \alpha_k\, d) - f(x_k) \;<\; -\rho(\alpha_k\, \|d\|), \tag{2.2}$$

  declare the iteration *successful*, set $x_{k+1} = x_k + \alpha_k\, d$ and skip the poll step.
  Otherwise, go to the poll step.
  Poll Step
  Compute a polling set D_k.
  If there exists $d \in D_k$ such that

$$f(x_k + \alpha_k\, d) - f(x_k) \;<\; -\rho(\alpha_k\, \|d\|), \tag{2.3}$$

  declare the iteration *successful* and set $x_{k+1} = x_k + \alpha_k\, d$.
  Otherwise, declare the iteration as *unsuccessful* and set $x_{k+1} = x_k$.
  Step Size Update
  If the iteration is successful, (possibly) increase the step size by setting $\alpha_{k+1} = \min\{\gamma\, \alpha_k, \alpha_{\max}\}$.
  Otherwise, decrease the step size by setting $\alpha_{k+1} = \theta\, \alpha_k$.
end

Secondly, the presence of $\alpha_{\max}$ is not necessary for establishing convergence; however, it is commonly adopted in practice, and a requirement to derive the upcoming complexity results. We thus chose to make it appear in the algorithm from the start.

Thirdly, the polling process performed in Algorithm 2.1 is *opportunistic* by default, i.e. we stop evaluating through the directions in D_k when we found one satisfying the sufficient decrease condition, in which case we move on to the next iteration.

Finally, note that practical implementations use a stopping criterion rather than a **for** loop, that often consists in specifying a maximum budget of function evaluations and/or a minimum tolerance for the step size. Relevance of the latter criterion will be discussed further in Section 2.2, and examples will be described in the numerical sections of this thesis.

2.1.3 Variations on the proposed framework

The above method covers two families of algorithms, namely those for which the forcing function ρ is equal to zero, and those for which $\rho(\alpha) > 0$ for $\alpha > 0$.

The second option corresponds to enforcing *sufficient decrease*, i.e., accepting a new point y if it decreases the function value by a certain amount measured through the function ρ

$$f(y) \ < \ f(x_k) - \rho(\alpha_k \, \|d_k\|). \tag{2.4}$$

It is one paradigm that, together with the use of Positive Spanning Sets, endows a directional direct-search algorithm with convergence properties. We will provide a detailed analysis of this setting in the rest of the chapter.

Several popular schemes rather rely on the first alternative, called *simple decrease*, meaning that the method will accept any new point that satisfies

$$f(y) \ < \ f(x_k), \tag{2.5}$$

where x_k is the current iterate. Such a decrease condition is the basis of the earliest direct-search methods, and the associated algorithms are often termed *Pattern Search methods*. Their convergence theory, essentially based on integer lattices and a finite number of polling sets, was first established by Torczon [109] in a smooth setting (assuming existence and Lipschitz continuity of the first-order derivative of the function f). An analysis for *Generalized Pattern Search* algorithms was then presented by Audet and Dennis [10], who derived a convergence analysis for various levels of smoothness of the objective function. The work in [10] can be considered as seminal for the later introduction of the *Mesh Adaptive Direct Search* (MADS) framework by the same authors [8]. The underlying analysis of this algorithm shows that the MADS-type methods generate sequences of iterates which possess guarantees at their limit points. For such schemes, global convergence to a first-order stationary point is ensured whenever the polling directions are asymptotically dense within the unit sphere, an assumption that can be satisfied deterministically [4].

Other similar approaches can be connected to Algorithm 2.1. Initiated by the work of Coope and Price, the Grid/Frame-based Search techniques (see [39] and references therein) have been successfully applied to both smooth and nonsmooth problems.

García-Palomares and Rodríguez [59] presented a framework derived from the pattern search framework but with sufficient decrease conditions, that was described in both sequential and parallel versions. It exploits the minimal frame theory of the Frame-based methods in order to guarantee convergence.

2.2 Global convergence theory

In this section, we derive classical global convergence proofs for Algorithm 2.1. Those proofs require the following assumptions on the objective function.

Assumption 2.1 *The function f is continuously differentiable on $\mathbb{R}^n$, and its gradient is Lipschitz continuous, of Lipschitz constant ν_g.*

Assumption 2.2 *The objective function f is bounded from below on its level set at x_0 given by $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$, and we denote by f_{low} a lower bound.*

A necessary assumption on the directions that are used is given below. Note that this property is required for for both the polling sets and the (possibly non-empty) sets of search directions.

Assumption 2.3 *There exists positive constants $d_{\min}$ and $d_{\max}$ such that every direction d that is considered for a poll step, or a search step when it is present, satisfies*

$$d_{\min} \leq \|d\| \leq d_{\max}.$$

A common way of satisfying Assumption 2.3 is by generating the polling sets so that they only contain unitary directions.

We end this series of assumptions by a requirement concerning the forcing function.

Assumption 2.4 *The forcing function is positive, non-decreasing on $(0, +\infty)$, and satisfies $\rho(\alpha) = o(\alpha)$ when $\alpha \to 0^+$.*

Standard choices for such a forcing function are $\rho(\alpha) = c\,\alpha^q$, with $c > 0,\ q > 1$. Note that our framework adopts the sufficient decrease approach.

2.2.1 Standard, liminf-type result

We begin this section by a useful lemma, that does not depend on the directions we use at every iteration (as long as those directions are bounded in norm).

Lemma 2.1 *Let Assumptions 2.2, 2.3 and 2.4 hold. Then, the step size sequence produced by Algorithm 2.1 satisfies*

$$\lim_{k \to \infty} \alpha_k = 0. \tag{2.6}$$

Proof. Suppose first that there are a finite number of successful iterations, and let k_0 denote the last index of those. Then, for every $k \geq k_0$, we have by the updating rule on the step size that

$$\alpha_k \;=\; \gamma\, \theta^{k-k_0-1}\, \alpha_{k_0},$$

that goes to zero when k goes to infinity, so in this case (2.6) holds.

Suppose now that there are infinitely many successful iterations. If there exists $\bar{\alpha} > 0$ such that $\forall\, k,\ \alpha_k > \bar{\alpha}$, then it also exists $\bar{\rho} > 0$ such that $\rho(\alpha_k \,\|d_k\|) \geq \rho(\alpha_k\, d_{\min}) > \bar{\rho}$ for all k. Let k be the index of a successful iteration; one has:

$$f(x_k) - f(x_{k+1}) \;>\; \rho(\alpha_k \,\|d_k\|) \;\;>\; \bar{\rho},$$

hence, by summing on all successful iterations and using Assumption 2.2, we arrive at

$$f_0 - f_{\text{low}} \;>\; \sum_{k \text{ successful}} \bar{\rho} \;=\; \infty. \tag{2.7}$$

We thus arrive at a contradiction, from which we deduce that $\liminf_k \alpha_k = 0$.

Suppose now that $\lim_k \alpha_k \neq 0$; then, for every $\epsilon > 0$, it exists a subsequence $\{k_i\}$ such that $\forall\, i,\ \alpha_{k_i} > \epsilon$. Since there are infinitely many successful iterations, we can assume without loss of generality that $\{k_i\}$ is a subsequence of successful iterations. As a result, by the same argument as before, it exists $\rho_\epsilon > 0$ such that for all i, $\rho(\alpha_{k_i}) > \rho_\epsilon$. Thus

$$f(x_{k_i}) - f(x_{k_{i+1}}) \;>\; f(x_{k_i}) - f(x_{k_i} + 1) \;>\; \rho(\alpha_{k_i}) \;>\; \rho_\epsilon,$$

and we obtain a contradiction similar to (2.7). □

Note that for some direct-search instances that are not based on sufficient decrease, the above result is only true for a subsequence of the step sizes in the general case. This still suffices to establish a convergence result. Note also that Lemma 2.1 does not assume any smoothness property on f, which makes this result an important one for practical purposes.

The next argument for our proof is related to the guarantees we have whenever an iteration is unsuccessful.

Lemma 2.2 *Suppose that Assumption 2.1 and 2.3 holds, and consider the index k of an unsuccessful iteration. Then, if D_k is a PSS, we have*

$$\|\nabla f(x_k)\| \;\leq\; \frac{1}{\text{cm}(D_k)} \left(\frac{\rho(\alpha_k\, d_{\max})}{\alpha_k\, d_{\min}} + \frac{\nu_g}{2} \alpha_k\, d_{\max} \right). \tag{2.8}$$

Proof. The proof is a classical result of the direct-search literature. If $\nabla f(x_k) = 0$, the result trivially holds, thus we assume in what follows that $\nabla f(x_k) \neq 0$. Since the iteration is unsuccessful, for each $d \in D_k$, one has

$$-\rho(\alpha_k) \;\leq\; f(x_k + \alpha_k\, d) - f(x_k). \tag{2.9}$$

Thus, a first-order Taylor expansion of $f(x_k + \alpha_k\, d)$ together with Assumption 6.1 leads to

$$\begin{aligned} -\rho(\alpha_k) &\leq \alpha_k\, \nabla f(x_k)^\top\, d + \alpha_k \int_0^1 \left[\nabla f(x_k + t\alpha_k\, d) - \nabla f(x_k)\right]^\top\, d\, dt \\ &\leq \alpha_k\, \nabla f(x_k)^\top\, d + \frac{\nu_g}{2}\, \alpha_k^2. \end{aligned}$$

Hence

$$-d^\top\, \nabla f(x_k) \;\leq\; \frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k. \tag{2.10}$$

Let d_k be a direction of D_k such that

$$-d_k^\top\, \nabla f(x_k) \;=\; \max_{d \in D_k} -d^\top\, \nabla f(x_k) \;\geq\; \mathrm{cm}(D_k)\, \|\nabla f(x_k)\|.$$

By considering this direction in (2.10) and using $\mathrm{cm}(D_k) > 0$, one obtains (2.8). □

Lemma 2.2 provides an upper bound the gradient norm on unsuccessful iterations. This bound goes to zero as α_k does, provided the sequence of cosine measures does not degenerate. The following assumption ensures that this is not the case.

Assumption 2.5 *It exists a constant $\kappa \in (0, 1)$ such that for any index k, $\mathrm{cm}(D_k) \geq \kappa$.*

Assumption 2.5 implies that at any iteration for which the current gradient is not zero, there will be a direction in D_k making an acute angle with the negative gradient, but an angle that does not get asymptotically close to $\frac{\pi}{2}$. This is critical even in derivative-based methods based on the use of descent directions [101].

We now have all the necessary ingredients to derive the standard convergence result of Algorithm 2.1, which is given in Theorem 2.1 (see [75, Theorem 3.11] for a proof).

Theorem 2.1 *Let Assumptions 2.1, 2.2, 2.4 and 2.5 hold. Then, the sequence of iterates generated by Algorithm 2.1 satisfies*

$$\liminf_{k \to +\infty} \|\nabla f(x_k)\| \;=\; 0.$$

One thus sees that the decrease of the gradient norm (on unsuccessful iterations) is a consequence of the step size going to zero. This motivates one of the usual stopping criteria in direct-search methods, which is

$$\alpha_k \;<\; \alpha_{tol}, \tag{2.11}$$

where α_{tol} is a positive tolerance. Thanks to Lemma 2.2, we know that whenever (2.11) holds at an unsuccessful iteration, then the gradient norm cannot be higher than a certain (albeit unknown) function of α_{tol}. This justifies further the choice of this criterion as an indicator of convergence.

2.2.2 A stronger, lim-type property

The results established in Section 2.2.1 can be strengthened by a change on the polling strategy. More precisely, we will assume that a *complete polling* is performed at every iteration, i.e. that the function is evaluated along all directions in D_k and only the point with the best function value is considered as a potential new iterate. This guarantees that any sufficient decrease that is accepted will be as important as the one obtained by the direction closest to the negative gradient.

In that case, we have the following additional result on successful iterations.

Proposition 2.3 *Suppose that we are in the assumptions of Theorem 2.1 and that complete polling is performed at every iteration. Then, for any successful iteration k of Algorithm 2.1 and any $\eta > 0$, there exists $\delta > 0$ and $\sigma > 0$ that do not depend on k such that if $\|\nabla f(x_k)\| > \eta$ and $\alpha_k < \delta$, then*

$$f(x_{k+1}) \;\leq\; f(x_k) - \sigma\,\alpha_k\,\|\nabla f(x_k)\|.$$

Proof. Let d denote a direction such that

$$-d^\top \nabla f(x_k) \;=\; \mathrm{cm}(D_k)\,\|d\|\,\|\nabla f(x_k)\| \;\geq\; \kappa\, d_{\min}\,\|\nabla f(x_k)\|.$$

As before, a Taylor expansion of $f(x_k + \alpha_k\, d) - f(x_k)$ yields

$$f(x_k + \alpha_k\, d) - f(x_k) \;\leq\; -\alpha_k\,\kappa\,\|\nabla f(x_k)\| + \frac{\nu_g}{2}\alpha_k^2.$$

Consequently, if we define $\delta = \frac{\kappa\, d_{\min}\,\eta}{\nu_g}$ and assume that $\alpha_k < \delta$, we arrive at

$$\begin{aligned} f(x_k + \alpha_k\, d) - f(x_k) &\leq -\alpha_k\,\kappa\, d_{\min}\,\|\nabla f(x_k)\| + \frac{\nu_g}{2}\,\alpha_k\,\frac{\kappa\, d_{\min}}{\nu_g}\eta && (2.12)\\ &\leq -\alpha_k\,\kappa\, d_{\min}\,\|\nabla f(x_k)\| + \frac{1}{2}\alpha_k\,\kappa\, d_{\min}\,\|\nabla f(x_k)\| && (2.13)\\ &= -\frac{\kappa\, d_{\min}}{2}\,\alpha_k\,\|\nabla f(x_k)\|. && (2.14)\end{aligned}$$

Because of the complete polling assumption, we know that the chosen direction d_k verifies

$$f(x_k + \alpha_k\, d_k) - f(x_k) \;\leq\; f(x_k + \alpha_k\, d) - f(x_k),$$

hence the result with $\sigma = \frac{\kappa\, d_{\min}}{2}$. □

This leads to a new convergence result presented in Theorem 2.2. The proof is a standard one, that can be found for instance in [75, Theorem 3.14].

Theorem 2.2 *Under the same assumptions as in Theorem 2.1, if complete polling is performed at every iteration, the sequence of iterates satisfies*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| \;=\; 0.$$

Note that the cost of a complete polling strategy can be prohibitive in practical applications if a single function evaluation is already expensive. For this reason, opportunistic polling is more often applied, despite its weaker theoretical guarantees.

2.3 Worst-case complexity

Complexity analysis has recently gained interest in the derivative-free community, following the outbreak of results for nonlinear nonconvex programming methods (see for instance [17, 24] and the references therein). In the case of direct search, Vicente [114] proved that a direct-search method based on sufficient decrease needs at most $\mathcal{O}\left(n\,\epsilon^{-2}\right)$ iterations to drive the gradient norm below some threshold $\epsilon \in (0,1)$. This result matches in terms of power of ϵ the iteration complexity bound obtained for steepest descent methods, with or without line search [22, 28]. It is also of the same order than classical optimization schemes such as standard trust-region [69] or Levenberg-Marquardt schemes [111]. Regularization methods have been shown to enjoy better complexity bounds [17, 26], a property shared by a recently proposed trust-region method [40].

In derivative-free optimization, the *evaluation complexity bounds* are also of interest, possibly even more than the iteration complexity ones. Indeed, since a derivative-free algorithm relies solely on the values of the objective function, its performance is often assessed based on its consumption of those function values. Even though the function may not be expensive to evaluate, the amount of evaluations needed by a method to reach a target value or to reduce the current value is generally a good performance indicator [95]. An asymptotic study of this cost might provide useful guidance in identifying the most economical (yet efficient) methods.

One can distinguish two types of results, that either look at the global decaying rate of the minimum gradient norm or estimate the number of iterations needed in the worst case to drive the norm of the gradient below a given tolerance. The resulting properties are termed *global rates* and *worst-case complexity bounds*, respectively.

2.3.1 Complexity analysis in the smooth, nonconvex case

In this section, we briefly state the arguments from [114] that allow for the derivation of a worst-case complexity bound. The idea is to focus first on the number of successful iterations, then to show that the number of unsuccessful ones is essentially of the same order than the number of successful ones. We will conduct a similar, detailed analysis in Chapter 6.

In [114], the author assumes that the forcing function is of the form $\rho(\alpha) = c\,\alpha^q$ with $c > 0$ and $q > 1$, which we will also do in the rest of this section. It is then shown that the choice $q = 2$ minimizes the complexity bound.

The general results are given below.

Theorem 2.3 *Let Assumptions 2.1, 2.2, 2.3 and 2.4 hold. Suppose further that the sequence of cosine measures* $\{\mathrm{cm}(D_k)\}$ *is bounded below by* $\kappa \in (0,1)$ *(therefore every* D_k *is a PSS).*

Then, the number of iterations needed by Algorithm 2.1 to reduce the gradient norm below some threshold $\epsilon \in (0,1)$ *is at most*

$$\mathcal{O}\left(\kappa^{-\frac{q}{\min(q-1,1)}}\,\epsilon^{-\frac{q}{\min(q-1,1)}}\right),$$

with the constant in $\mathcal{O}(\cdot)$ depending on $\alpha_0, f(x_0), f_{\text{low}}, \nu_g, d_{\min}, d_{\max}, \theta, \gamma, c, q$.

This bound is minimal when $q = 2$, in which case the bound is $\mathcal{O}\left(\kappa^{-2}\,\epsilon^{-2}\right)$.

In classical derivative-based methods, the iteration complexity can be directly associated with the evaluation complexity, that is, there exist bounds on the number of derivative or function evaluations that follows from the bound on the number of iterations. The bounds often correspond one-to-one: for instance, in a basic trust-region method, only one evaluation of the function, gradient and Hessian are required per iteration. In derivative-free optimization, one can also obtain results regarding the number of function evaluations needed by the method to satisfy an approximate form of stationarity. Such bounds are of critical importance, since they represent a (worst-case) performance indicator of the consumption of the algorithm.

In the case of direct search, the result derived from Theorem 2.3 is the following.

Corollary 2.1 *Under the assumptions of Theorem 2.3, the number of function evaluations needed by Algorithm 2.1 to drive the gradient below $\epsilon \in (0, 1)$ is*

$$\mathcal{O}\left(m\,\kappa^{-\frac{q}{\min(q-1,1)}}\,\epsilon^{-\frac{q}{\min(q-1,1)}}\right),$$

with the constant in $\mathcal{O}(\cdot)$ depending on $\alpha_0, f(x_0), f_{\text{low}}, \nu_g, d_{\min}, d_{\max}, \theta, \gamma, c, q$.

Again, this bound is minimized by the choice $q = 2$, in which case it becomes

$$\mathcal{O}\left(m\,\kappa^{-2}\,\epsilon^{-2}\right).$$

A classical polling choice is the coordinate set $D_k = D_{\oplus}$; in that case, we have $m = 2\,n$ and $\kappa = \frac{1}{\sqrt{n}}$, so the above bounds become $\mathcal{O}(n\,\epsilon^{-2})$ and $\mathcal{O}(n^2\,\epsilon^{-2})$, respectively.

Dodangeh, Vicente and Zhang have shown that such a dependence in n is optimal for direct-search methods based on Positive Spanning Sets [48]. Regarding the dependence on ϵ, Cartis, Gould and Toint [26] have developed a derivative-free method with a complexity bound in $\epsilon^{-3/2}$ (note that this does not imply that the bound from Theorem 2.3 is always worse, see the last section of [26] for a discussion on this topic). Despite this result, it can be argued that direct-search methods have more in common with steepest descent methods, for which ϵ^{-2} is the optimal power of ϵ that can be obtained. Vicente [114] remarks that the pathological example from [22] in dimension 1 can also serve in the derivative-free case to demonstrate the sharpness of the ϵ^{-2} for smooth nonconvex functions.

2.3.2 Other complexity bounds

The bounds established in Section 2.3.1 are valid for any smooth, nonconvex optimization problem. Dodangeh and Vicente [47] provided a iteration complexity bound for Algorithm 2.1 in $\mathcal{O}\left(n\,\epsilon^{-1}\right)$ in the case of a convex objective, and $\mathcal{O}\left(n\,|\log\epsilon|\right)$ for a strongly convex objective (see also Konečný and Richtárik [78] in a simplified setting). Interestingly, numerical experimentation presented in [47] showed that a direct-search

scheme based upon Algorithm 2.1 outperformed methods that were tailored for convex optimization problems [53, 99], even though they presented similar or improved complexity bounds in terms of power of the tolerance ϵ.

In the nonsmooth case, it was shown by Garmanjani and Vicente [61] that a smoothing direct-search method (minimizing a sequence of smooth problems converging to the nonsmooth one) would require at most $\mathcal{O}\left(n \left|\log \epsilon\right| \epsilon^{-3}\right)$ iterations to drive the smoothing parameter under ϵ.

Given that the convergence analyses in the (strongly) convex and nonsmooth cases are also based on PSSs, the evaluation complexity bound corresponds to the iteration complexity increased by a factor of the order of the dimension n, thus resulting in a total dependence in the square of the dimension. As a result, the dependence on the dimension for the number of function evaluations is not altered by the smoothness properties of the objective function.

2.4 Other types of direct-search methods

In this section, we briefly cover other main classes of direct-search algorithms, stressing out the common aspects between these classes and the one at the heart of our study.

2.4.1 Line-search methods based on positive spanning sets

Line search is one technique that can be used to ensure global convergence of optimization methods such as the steepest descent algorithm or Newton's method [101]. In a derivative-free setting, the line search has the same function but must be coupled with another guarantee (search along directions obtained either by a PSS or through a model) in order to arrive at a convergence result.

We remark that in the literature, those methods are more often referred to as *derivative-free line-search* methods rather than direct-search ones, the latter being precisely used to emphasize the absence of line search. We will adopt a similar terminology in the rest of the thesis.

A typical derivative-free line-search method relies on iteratively evaluating the objective function on a set of directions, aiming for sufficient decrease (of the order of α^2, where α is the step size). Thus, those methods initially behave like the ones we presented in Section 2.1, but they look deeper into successful steps. Indeed, when a step producing a sufficient decrease has been identified, a line search is performed to look for improvement with larger (yet not arbitrarily) step sizes. In a deterministic setting, such a strategy is usually applied with a fixed set of directions with associated step sizes.

For line-search algorithms relying on directions originating from a PSS, the convergence results were actually established prior to those for directional direct-search without line search [70]. An analysis covering both classes of methods was then developed by Lucidi and Sciandrone [88]. Those results enlighten the closeness between the two approaches. Still, they may give rise to optimization methods with significantly different consumptions of function evaluations and, consequently, distinct practical performances.

2.4.2 Directional direct search based on simplex derivatives

The use of *simplex* derivatives as indicators of promising search directions has been investigated in the directional direct-search setting, both with and without performing a line search on such vectors.

Definition 2.4 *Let $Y = \{y^0, \ldots, y^m\}$ be a set of* affine independent points *in $\mathbb{R}^n$, i.e. such that its affine hull*

$$\text{aff}\,(Y) \;=\; \left\{ \sum_{i=0}^{m} \beta_i\, y_i \;\middle|\; \sum_{i=0}^{m} \beta_i = 1 \right\}$$

has dimension m.

The convex hull of Y, defined by

$$\text{conv}\,(Y) \;=\; \left\{ \sum_{i=0}^{m} \beta_i\, y_i \;\middle|\; \beta_i \geq 0,\; \sum_{i=0}^{m} \beta_i = 1 \right\}$$

is called a simplex of dimension m. *The vectors $y^0, \ldots, y^m$ are called the* vertices *of the simplex.*

Simplices are helpful in defining approximations of the derivatives of the function f when those exist.

Definition 2.5 *Let $Y = \{y^0 \cdots y^m\}$ be a set of points in $\mathbb{R}^n$ and assume that the matrix*

$$L_Y \;=\; \left[y^1 - y^0 \;\; \cdots \;\; y^m - y^0 \right]^\top$$

is of full rank, i.e. $\text{rank}(L_Y) = \min\{n, m\}$. *Then, the* simplex gradient *of f at y^0 is defined as the solution of the system*

$$L_Y\, \nabla_s f\left(y^0\right) \;=\; \delta(f, Y). \tag{2.15}$$

where $\delta(f, Y) = \left[f(y^1) - f(y^0), \ldots, f(y^m) - f(y^0) \right]^\top$.

Whenever $m > n$, the system is solved in the least-squares sense; in that situation, the simplex gradient is sometimes referred to as the *stencil gradient* [74].

The implicit filtering method [74] is a direct-search method based on an approximate quasi-Newton direction, computed from the simplex (or stencil) derivatives established using a positive basis (typically $D_\oplus$). Contrary to the previous section, the line search performed here is a backtracking process, that looks for a step size satisfying a simple decrease in the function values. The stencil gradient is also used as a measure of stationarity to terminate the algorithm.

Custódio and Vicente considered a pattern search method [42] in which a simplex gradient was built using previous function evaluations. This vector is not directly employed as a poll direction, but is used as a descent indicator to order the directions of

the polling sets so that the closest to this gradient estimate is considered first in the poll step. The resulting methods were shown to consume less function evaluations in practice. Following this study, Custódio proposed two algorithms based on simplex derivatives to determine descent directions, one being an hybridization of backtracking line search and direct search based on sufficient decrease [41, Chapter 7].

2.4.3 Simplicial direct-search methods

So far we have presented *directional* direct-search algorithms, that form an important subclass of the direct search one as they encompass all the frameworks we described in the previous sections. *Simplicial* direct-search methods represent another direct-search subclass that focuses on using simplices rather than polling sets while searching for an optimal solution. One significant difference with the approaches presented above is that the simplicial algorithms typically move away from the worst known point instead of moving to the best point. This process is justified by the use of a simplex at every iteration, supposedly helpful in capturing information about the geometry.

One of the most famous direct-search methods of simplicial type (and one of the most popular optimization methods) is the Nelder-Mead algorithm [98]. This algorithm proceeds by evaluating the function at each iteration on a simplex, adapting this simplex afterwards in order to approximate the curvature of the function. One inconvenient of the use of simplices is that the simplex sequence generated by the method may yield arbitrary flat or needle-like shapes. This generally leads to stagnation of the algorithm, thus preventing the iterate sequence from converging. Such behavior can be observed on simple two-dimensional examples (note that global convergence is guaranteed in dimension 1).

Despite its lack of general convergence analysis, the Nelder-Mead algorithm has been successfully applied on a wide range of problems (see [38, Chapter 8] and references therein). Besides, globally convergent variants on the Nelder-Mead framework have been developed, such as the variants based on sufficient decrease proposed by Tseng [110] or the Multidirectional search (MDS) method [44].

Although the Nelder-Mead algorithm is known to fail to converge on simple problems, it is still a method of choice for practitioners, due to its simplicity of implementation and testing. It is interesting to note that the same features are shared by most methods from the direct-search family.

2.5 Conclusion and references for Chapter 2

Direct-search methods have been extensively studied in derivative-free optimization, and ever since their introduction, remain a popular choice of algorithms to tackle problems for which derivatives cannot be obtained. Thanks to their easily understandable and implementable paradigms, they can often serve as a first attempt to address a minimization problem. When simple strategies are adopted to choose the directions used for exploration of the space, it is possible to derive a convergence as well as a complexity study

of those frameworks. Such rather recent advances have contributed to distinguish this class of methods from pure heuristics: consequently, direct search also regained interest from a theoretical point of view.

We presented a directional direct-search framework to serve as a reference basis in the rest of the thesis. For such a simple scheme, both convergence and complexity results hold in a smooth setting. We identified the main arguments that intervene in the related proofs, and provided examples of polling strategies that satisfy the requirements of the convergence theory.

For a thorough description of the directional direct-search family of methods, we refer to the review paper by Kolda, Lewis and Torczon [75] and references therein. Details on other classes of direct-search algorithms, as well as descriptions of the approximation tools, can be found in the textbook by Conn, Scheinberg and Vicente [38], where the direct-search class is presented in the larger context of deterministic derivative-free optimization.

Nota Bene: *In the rest of the thesis, by a direct-search algorithm we will always mean one of directional type, and, unless explicitly stated otherwise, its algorithmic framework will follow the template given in Section 2.1.*

Chapter 3

Direct search using probabilistic descent

Direct-search methods generally suffer from an increase in the dimensionality of the problems that are to be solved. One of the explanations for such a behavior lies in the enforced requirements on the polling directions. Indeed, the classical convergence analysis involves the use of Positive Spanning Sets, which implies that at least $n + 1$ directions are considered at every iteration in dimension n. As this number grows linearly with the dimension, the cost of a direct-search run may become prohibitive; in the worst case, the amount of evaluations needed in order to perform the (approximate) minimization of an objective function may even increase quadratically with respect to the dimension.

This chapter describes a probabilistic approach that is able to reduce this dependence on the dimension, by certifying quality of the polling sets only in probability. After replacing this study in its context in Section 3.1, we describe and motivate the proposed algorithmic framework in Section 3.2. Section 3.3 provides a convergence analysis of the method, distinguishing between the results that follow from the deterministic case, and those that are obtained using tools from probability theory. A complexity analysis is derived in Section 3.4, which illustrates the potential benefits of the probabilistic technique. Practical satisfaction of the desired properties is guaranteed through a distribution of the directions detailed in Section 3.5. The related results are then put in perspective with the performance of a practical implementation in Section 3.6.

3.1 Probabilistic quantities in derivative-free optimization

Rather than developing algorithms that fall into the category of stochastic DFO methods, our goal is to design algorithms that lie in between the deterministic and stochastic classes. To this end, we propose the following definition and terminology. Our goal is to propose a method that originates from a deterministic framework, in which one key component of the theoretical analysis is turned into a random variable or process. We emphasize the fact that such an algorithm will have a deterministic counterpart with (in particular) a well-established convergence theory. We expect to be able to maintain the associated properties, while improving the practical performance of the method.

To this respect, the work of Bandeira, Scheinberg and Vicente [14] can be seen as seminal, since it contains a convergence analysis of a derivative-free trust-region method based on random models. The arguments in [14] were then adapted in other trust-region contexts, essentially addressing noisy or stochastic problems [30, 79] (see also [15] for an extension to a Levenberg-Marquardt algorithm). The contribution presented in this chapter [67] also used the same tool to prove convergence for a direct-search scheme, then proposed a technique for establishing a probabilistic worst-case complexity bound, which is applicable to the trust-region framework of [14]. More recently, Cartis and Scheinberg [29] have proposed a different complexity analysis for line-search and cubic regularization methods based on probabilistic models, thereby also establishing convergence properties for the corresponding algorithms.

3.2 A direct-search scheme based on probabilistic descent directions

We present a variant on the direct-search framework of Algorithm 2.1, where randomness is introduced at every iteration by means of randomly generated polling sets, in an independent fashion. At first glance, such a variation is only impacting the reproducibility of the results, and does not necessarily represent any improvement. However, we present experimentations produced by the execution of both deterministic and random variants in MATLAB [90] that encourage using considerably less directions that what is imposed in the deterministic analysis of Chapter 2.

3.2.1 Algorithm

Algorithm 3.1 gives the formal statement of our method. Note that we removed the search step, since it does not interfere in our theoretical analysis. A comparison based solely on poll steps will allow us to study the impact of our new probabilistic polling strategy. Note, however, that the upcoming analysis still holds even if a search step is included in Algorithm 3.1, and possibly taken. The analysis would then consider $\mathfrak{D}_k$ defined at all iterations even if the poll step is skipped.

Apart from that change, one can see that Algorithm 3.1 relies on the same ideas as Algorithm 2.1, only with randomly generated polling sets. This not only implies that

Algorithm 3.1: Direct Search method based on Probabilistic Descent (DSPD)

Choose an initial point $x_0 \in \mathbb{R}^n$, as well as $0 < \theta < 1 \leq \gamma$, $0 < \alpha_0 < \alpha_{\max}$.
Define a *forcing function* $\rho : \mathbb{R}^+ \to \mathbb{R}^+$.
for $k = 0, 1, 2, \ldots$ **do**
  Poll Step
  Compute a set of random independent vectors $\mathfrak{D}_k$.
  If there exist $\mathfrak{d} \in \mathfrak{D}_k$ such that

$$f(X_k + \mathcal{A}_k\,\mathfrak{d}) - f(X_k) \;<\; -\rho(\mathcal{A}_k\,\|\mathfrak{d}\|),$$

  declare the iteration *successful* and set $X_{k+1} = X_k + \mathcal{A}_k\,\mathfrak{d}$;
  Otherwise declare the iteration as *unsuccessful* and set $x_{k+1} = x_k$.
  Step Size Update
  If the iteration is successful, (possibly) increase the step size by setting $\mathcal{A}_{k+1} = \min\{\gamma\,\mathcal{A}_k, \alpha_{\max}\}$;
  Otherwise, decrease the step size by setting $\mathcal{A}_{k+1} = \theta\,\mathcal{A}_k$.
end

the polling sets and directions are sets of random vectors, but it also means that every iterate and step size (save for the initial ones) is a random vector or variable, respectively. Table 3.1 summarizes the relevant notations.

Element	Random notation	Realization notation
Polling set	$\mathfrak{D}_k$	D_k
Step size	$\mathcal{A}_k$	α_k
Iterate	X_k	x_k

Table 3.1: Notations for direct search with randomly generated directions.

3.2.2 A preliminary numerical illustration

The motivation behind Algorithm 3.1 is the possible gain that one may achieve by using directions that were generated in a random fashion. To better identify this potential improvement, we conducted a preliminary experiment by running a MATLAB implementation of Algorithm 3.1 for different choices (deterministic and random) of the polling directions. The test problems were taken from the `CUTEr` [64] collection (now available as `CUTEst` [65]) with dimension $n = 40$. We set $\alpha_{\max} = \text{Inf}$, $\alpha_0 = 1$, $\theta = 0.5$, $\gamma = 2$, and $\rho(\alpha) = 10^{-3}\alpha^2$. The algorithm terminated when either α_k was below a tolerance of 10^{-10} or a budget of $2000n$ function evaluations was exhausted.

The first three columns of Table 3.2 correspond to three PSS choices for the polling directions D_k: $D_\oplus$ represents the columns of the identity matrix of size n; $Q\,D_\oplus$ means

Table 3.2: Relative performance for different sets of polling directions ($n = 40$).

	$D_\oplus$	$Q\,D_\oplus$	$Q_k\,D_\oplus$	$2n$	$n+1$	$n/2$	$n/4$	2	1
arglina	3.42	8.44	16.67	10.30	6.01	3.21	1.88	1.00	–
arglinb	20.50	10.35	11.38	7.38	2.81	2.35	1.85	1.00	2.04
broydn3d	4.33	6.55	11.22	6.54	3.59	2.04	1.28	1.00	–
dqrtic	7.16	9.37	19.50	9.10	4.56	2.77	1.70	1.00	–
engval1	10.53	20.89	23.96	11.90	6.48	3.55	2.08	1.00	2.08
freuroth	56.00	6.33	1.33	1.00	1.67	1.33	1.67	1.00	4.00
integreq	16.04	16.29	18.85	12.44	6.76	3.52	2.04	1.00	–
nondquar	6.90	30.23	17.36	7.56	4.23	2.76	1.87	1.00	–
sinquad	–	–	2.12	1.65	2.01	1.26	1.00	1.55	–
vardim	1.00	3.80	3.30	1.80	2.40	2.30	1.80	1.80	4.30

the columns of Q and $-Q$, where Q is an orthogonal matrix obtained by the QR decomposition of a random vector, uniformly distributed on the unit sphere of $\mathbb{R}^n$, generated before the start of the algorithm and then fixed throughout the iterations; $Q_k\,D_\oplus$ consists of the columns of Q_k and $-Q_k$, where Q_k is a matrix obtained in the same way as Q except that it is generated independently at each iteration. We point out that although some of those sets are randomly generated at every iteration, they always form a PSS, thus the *deterministic* analysis of Chapter 2 holds. This illustrates that a randomized approach does not necessarily require a probabilistic analysis. In the cases of $D_\oplus$ and $Q\,D_\oplus$, a cyclic polling procedure was applied to accelerate descent, by starting polling at the direction that led to previous success (if the last iteration was successful) or at the direction right after the last one used (if the last iteration was unsuccessful). This technique has been shown to improve the performance of (deterministic) direct-search methods [42].

In the remaining columns of Table 3.2, $\mathfrak{D}_k$ is a random set (hence the change of notation) consisting of m independent random vectors uniformly distributed on the unit sphere in $\mathbb{R}^n$, with $m = 2n, n+1, n/2, n/4, 2, 1$. Those vectors are independently generated at every iteration.

For every polling choice, we counted the number of function evaluations taken to drive the function value below $f_{\text{low}} + \epsilon\,[f(x_0) - f_{\text{low}}]$, where f_{low} is the true minimal value of the objective function f, and ϵ is a tolerance set to 10^{-3}. Given a test problem, we present in each row the ratio between the number of function evaluations taken by the corresponding version and the number of function evaluations taken by the best version. Because of the random nature of the computations, the number of function evaluations was obtained by averaging over ten independent runs, except for the first column. The symbol '–' indicates that the algorithm failed to solve the problem to the required precision at least once in the ten runs.

When $\mathfrak{D}_k$ is randomly generated on the unit sphere (columns $m = 2n, n+1, n/2, n/4$, $2, 1$ in Table 3.2), we know that the vectors in $\mathfrak{D}_k$ do not form a PSS when $m \leq n$, and there is no guarantee that they do when $m > n$. However, we can clearly see from Table 3.2 that randomly generating the polling directions in this way performed better, despite the fact that their use is not covered by the classical theory of direct search. This evidence motivates the derivation of a convergence theory of Algorithm 3.1 covering the case when the polling directions are randomly generated.

3.3 Global convergence theorems

In what follows, we consider the random process generating the polling directions of Algorithm 3.1. Our objective is to identify properties of those directions that can ensure convergence of the framework, hopefully reducing to the deterministic assumptions of Section 2.2 when the sets are actually deterministically generated.

When performing a probabilistic study of an algorithm, one can obtain several types of results, holding with various levels of certainty. We identify thereafter a hierarchy of such properties.

A result holding *for any realization* of an algorithm is a result that does not involve arguments sensitive to the random nature of the method. It essentially holds whether we perform a deterministic or a probabilistic analysis of the algorithm.

A property established *almost surely* or *with probability one* holds except for a set of realizations that can be neglected (of probability measure equal to zero). It often is a consequence of a theoretical probability result that can be stated with probability one. Note that if a property holds for all realizations (i.e., deterministically), then it holds almost surely.

Finally, the weakest type of results we will consider will only be guaranteed to hold *with a minimum probability*. Again, one may notice that an almost-sure result (and, consequently, one holding for any realization) holds with any minimum probability between 0 and 1.

3.3.1 Preliminary results at the realization level

A typical convergence proof in direct search relies on two elements: the convergence of (at least) a step size subsequence to zero, and a property linking the polling directions and the negative gradient on a related subsequence.

In Section 2.2, we saw that the proof of $\alpha_k \to 0$ did not require any assumption on the polling sets [75]. A similar result can thus be derived by applying exactly the same argument to a realization of direct search now based on probabilistic descent, which yields the following lemma.

Lemma 3.1 *For each realization of Algorithm 3.1,* $\lim_{k\to\infty} \alpha_k = 0$.

Since we aim at generating sets with less than $n + 1$ directions, we know that those sets cannot positively span the variable space. However, one can take a closer look at

the positive spanning property and identify the part of this property that is relevant for a convergence proof.

Definition 3.1 *Consider a set of (nonzero) vectors $D \subset \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$. Let $\kappa \in (0,1)$; the set D is called a* κ-descent set at v *if*

$$\mathrm{cm}(D,v) \;=\; \max_{d\in D} \frac{d^\top v}{\|d\|\|v\|} \;\geq \kappa, \tag{3.1}$$

where we set $\mathrm{cm}(D,0) = 1$.

The quantity $\mathrm{cm}(D,v)$ is called the *cosine measure of D at v*. It is directly related to the cosine measure we presented in Section 2.1.1 by

$$\mathrm{cm}(D) \;=\; \min_{\|v\|\neq 0} \mathrm{cm}(D,v).$$

(Note that we may also include the case $v = 0$.)

One thus sees a PSS D is also a κ-descent set for *any nonzero vector in the space* and any $\kappa \leq \mathrm{cm}(D)$. On the other hand, for a specific v, there exist sets of vectors that are not PSSs and yet satisfy (3.1) for some κ (the simplest example being $D = \{v\}$).

In the rest of this chapter, we will make the following simplifications.

Assumption 3.1 *For any $k \geq 0$, the polling set D_k is formed by unitary vectors.*

Lemma 3.2 illustrates why the κ-descent property is as suitable as using Positive Spanning Sets. It can be proven the same way as Lemma 2.2.

Lemma 3.2 *Let Assumption 2.1 and 3.1 hold, and consider the k-th iteration of a realization of Algorithm 3.1. Let $g_k = \nabla f(x_k)$ and suppose that D_k is a κ-descent set at $-g_k$, satisfying Assumption 2.3. Then,*

$$\|g_k\| \;\leq\; \kappa^{-1}\left[\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k\right]. \tag{3.2}$$

Therefore, using κ-descent sets, it is possible to arrive at the same convergence analysis than in the case of deterministic direct search based on Positive Spanning Sets.

In order to make the algebra more concise, we will use a simplified notation for (3.2). For each $t > 0$, the value

$$\varphi(t) \;=\; \inf\left\{\alpha : \alpha > 0,\; \frac{\rho(\alpha)}{\alpha} + \frac{\nu_g}{2}\,\alpha \;\geq\; t\right\} \tag{3.3}$$

is well defined, and the set of all values defines a non-decreasing function φ. Note that Assumption 2.4 ensures that $\varphi(t) > 0$ for $t > 0$. When $\rho(\alpha) = c\,\alpha^2/2$, for instance, one obtains $\varphi(t) = 2t/(c + \nu_g)$.

Using this function, we can then reformulate Lemma 3.2 as follows.

Lemma 3.3 *Consider a realization of Algorithm 3.1 under the assumptions of Lemma 3.2. Its k-th iteration is successful if*

$$\mathrm{cm}(D_k, -g_k) \;\geq\; \kappa \quad \textit{and} \quad \alpha_k \;<\; \varphi(\kappa\|g_k\|).$$

Proof. According to the definition of $\mathrm{cm}(D_k, -g_k)$, there exists $d_k^* \in D_k$ satisfying

$$d_k^{*\top} g_k \;=\; -\,\mathrm{cm}(D_k, -g_k)\|d_k^*\|\|g_k\| \;\leq\; -\kappa\|g_k\|.$$

Thus, by Taylor expansion,

$$f(x_k + \alpha_k\, d_k^*) - f(x_k) \;\leq\; \alpha_k\, d_k^{*\top} g_k + \frac{\nu_g}{2}\,\alpha_k^2 \;\leq\; -\kappa\,\alpha_k\,\|g_k\| + \frac{\nu_g}{2}\,\alpha_k^2.$$

Using the definition of φ, we obtain from $\alpha_k < \varphi(\kappa\|g_k\|)$ that

$$\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\,\alpha_k \;<\; \kappa\|g_k\|,$$

hence $f(x_k + \alpha_k\, d_k^*) < f(x_k) - \rho(\alpha_k)$, and thus the k-th iteration is successful. □

This reformulation will be of use for both the convergence results and the complexity analysis of Section 3.4.

So far we have established deterministic results, holding for all realizations. Although Lemma 3.1 does not depend on the way we choose the polling sets, the other results involve the κ-descent property, which may not be satisfied if we generate the directions at random. In order to obtain a convergent behavior, we will need to consider sets that satisfy this property "sufficiently often" for the method to eventually provide the same guarantees as in the deterministic case.

3.3.2 A submartingale and its associated property

The following concept of probabilistically descent sets of polling directions is critical to our analysis. We use it to describe the quality of the random polling directions in Algorithm 3.1. Recalling that $g_k = \nabla f(x_k)$, let G_k be the random variable corresponding to g_k.

Definition 3.2 *The sequence $\{\mathfrak{D}_k\}$ in Algorithm 3.1 is said to be* p-probabilistically κ-descent*, or (p,κ)-descent in short, if*

$$\mathbb{P}\left(\mathrm{cm}(\mathfrak{D}_0, -G_0) \;\geq\; \kappa\right) \;\geq\; p$$

and, for each $k \geq 1$,

$$\mathbb{P}\left(\,\mathrm{cm}(\mathfrak{D}_k, -G_k) \;\geq\; \kappa \mid \sigma\left(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right)\right) \;\geq\; p, \tag{3.4}$$

where $\sigma\left(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right)$ is the σ-algebra generated by the random sets $\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}$.

Definition 3.2 requires the cosine measure $\mathrm{cm}(\mathfrak{D}_k, -G_k)$ to be favorable in a probabilistic sense rather than deterministically. It was inspired by the definition of probabilistically fully linear models (to be described in Chapter 4). Inequality (3.4) involves the notion of conditional probability (see [106, Chapter II]) and says essentially that the probability of the event $\{\mathrm{cm}(\mathfrak{D}_k, -G_k) \geq \kappa\}$ is not smaller than p, no matter what happened with $\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}$. It is stronger than merely assuming that $\mathbb{P}\left(\mathrm{cm}(\mathfrak{D}_k, -G_k) \geq \kappa\right) \geq p$.

For every index $k \geq 0$, we now define Y_k as the indicator function of the event

$$\{\text{the } k\text{-th iteration is successful}\}.$$

Furthermore, Z_k will be the indicator function of the event

$$\{\mathrm{cm}(\mathfrak{D}_k, -G_k) \geq \kappa\},$$

where we point out that $\kappa \in (0, 1)$ is independent of the iteration counter. The Bernoulli processes $\{Y_k\}$ and $\{Z_k\}$ will play a major role in our analysis. Their realizations are denoted by $\{y_k\}$ and $\{z_k\}$. Notice, then, that Lemma 3.3 can be restated as follows: given a realization of Algorithm 3.1 and $k \geq 0$, if $\alpha_k < \varphi(\kappa\|g_k\|)$, then $y_k \geq z_k$.

Lemmas 3.1 and 3.3 lead to a critical observation presented below as Lemma 3.4. We observe that such a result holds without any assumption on the probabilistic behavior of $\{\mathfrak{D}_k\}$.

Lemma 3.4 *For the stochastic processes $\{G_k\}$ and $\{Z_k\}$, where $G_k = \nabla f(X_k)$ and Z_k is the indicator of the event of $\mathfrak{D}_k$ being κ-descent, it holds that*

$$\left\{\liminf_{k\to\infty} \|G_k\| \;>\; 0\right\} \;\subset\; \left\{\sum_{k=0}^{\infty} \left[Z_k \ln\gamma + (1 - Z_k)\ln\theta\right] \;=\; -\infty\right\}. \tag{3.5}$$

Proof. Consider a realization of Algorithm 3.1 for which $\liminf_{k\to\infty} \|g_k\|$ is not zero but a positive number ϵ. There exists a positive integer k_0 such that for each $k \geq k_0$ it holds $\|g_k\| \geq \epsilon/2$ and $\alpha_k < \varphi(\kappa\epsilon/2)$ (because $\alpha_k \to 0$ and $\varphi(\kappa\epsilon/2) > 0$), and consequently $\alpha_k < \varphi(\kappa\|g_k\|)$. Hence we can obtain from Lemma 3.3 that $y_k \geq z_k$. Additionally, we can assume that k_0 is large enough to ensure $\alpha_k \leq \gamma^{-1}\alpha_{\max}$ for each $k \geq k_0$. Then the stepsize update of Algorithm 3.1 gives us

$$\alpha_k \;=\; \alpha_{k_0} \prod_{l=k_0}^{k-1} \left(\gamma^{y_l}\theta^{1-y_l}\right) \;\geq\; \alpha_{k_0} \prod_{l=k_0}^{k-1} \left(\gamma^{z_l}\theta^{1-z_l}\right)$$

for all $k \geq k_0$. This leads to $\prod_{l=0}^{\infty} \left(\gamma^{z_l}\theta^{1-z_l}\right) = 0$, since $\alpha_{k_0} > 0$ and $\alpha_k \to 0$. Taking logarithms, we conclude that

$$\sum_{l=0}^{\infty} \left[z_l \ln\gamma + (1 - z_l)\ln\theta\right] \;=\; -\infty,$$

which completes the proof. □

Given Lemma 3.4, we need only prove that the right-side event of (3.5) has a probability zero of occurrence. This is possible under the following assumption.

Assumption 3.2 *The polling set sequence is a (p, κ)-descent sequence with $p \geq p_0$, where*

$$p_0 \;=\; \frac{\ln \theta}{\ln(\gamma^{-1}\theta)}. \tag{3.6}$$

Assumption 3.2 means that the probability of the variable Z_k to be equal to 1, no matter what happened for $Z_0, \ldots, Z_{k-1}$, is higher than p_0. It can then be checked that the random process

$$\left\{ \sum_{l=0}^{k} \left[Z_l \ln \gamma + (1 - Z_l) \ln \theta \right] \right\}$$

is a submartingale with bounded increments (see Appendix A for a precise definition).

Proposition 3.1 *Let Assumption 3.2 hold. The sequence of random variables*

$$W_k \;=\; \sum_{l=0}^{k} \left[Z_l \ln \gamma + (1 - Z_l) \ln \theta \right]$$

is a submartingale for the filtration $\{\sigma_k = \sigma(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1})\}_k$, with bounded increments.

Proof. It is clear that the sequence $\{W_k\}$ has bounded increments. Moreover, by definition, W_k is σ_{k-1}-measurable and $\mathbb{E}|W_k| < \infty$. Finally, we have that

$$\begin{aligned}
\mathbb{E}\left[W_k | \sigma_{k-1}\right] &= \mathbb{E}\left[W_{k-1} | \sigma_{k-1}\right] + (\ln \gamma - \ln \theta)\, \mathbb{E}\left[Z_k | \sigma_{k-1}\right] + \ln \theta \\
&= W_{k-1} + (\ln \gamma - \ln \theta)\, \mathbb{P}\left(Z_k = 1 | \sigma_{k-1}\right) + \ln \theta \\
&\geq W_{k-1} + (\ln \gamma - \ln \theta)\, p - \ln \theta \\
&\geq W_{k-1},
\end{aligned}$$

using Assumption 3.2. □

From Theorem 5.3.1 and Exemple 5.3.1 of [54], we thus have that

$$\mathbb{P}\left[\, W_k \to c \in \mathbb{R} \,\cup\, \limsup W_k = \infty\right] = 1.$$

As the increment of W_k is either $\ln \gamma$ or $\ln \theta$, it cannot converge (as it cannot remain arbitrary close to any real value), which implies that the second event happens with probability 1, i.e.,

$$\mathbb{P}\left(\limsup W_k \;=\; \infty\right) \;=\; 1. \tag{3.7}$$

Hence the event on the right-hand side of (3.5) has probability zero.

3.3.3 Main convergence theorem

In the previous sections, we detailed all the necessary ingredients to prove a convergence result for Algorithm 3.1, in a similar fashion than [14] for the case of trust-region methods based on probabilistic models. We are now ready to state our main convergence result.

Theorem 3.1 *Let Assumptions 2.1, 2.2, 2.4, 3.1 and 3.2 hold. Then, the sequence of iterates produced by Algorithm 3.1 satisfies*

$$\mathbb{P}\left(\liminf_{k\to\infty} \|\nabla f(X_k)\| = 0 \right) \;=\; 1. \tag{3.8}$$

Theorem 3.1 shows a convergence result that is *almost* as strong as the deterministic one, in the sense that the set of realizations of Algorithm 3.1 for which it does not hold is of measure zero. We can thus say that almost every run of the method will converge to a first-order stationary point.

We point out that this lim inf-type global convergence result is also implied by the global rate theory of the next section, under a slightly stronger assumption (see [67] for details).

3.3.4 A stronger convergence result

We now extend the convergence analysis derived in [67]. Similarly to the deterministic case, we can show that the whole sequence of gradient norms goes to zero. The proof follows the reasoning of the trust-region case [14], and requires the following additional assumption.

Assumption 3.3 *The polling in Algorithm 3.1 is complete, meaning we perform all m function evaluations at each iteration.*

With the complete polling assumption, we know that whenever the polling set satisfies the κ-descent property, the decrease can be expressed as a function of the gradient norm. This is the statement of the following proposition, which is a variation on Proposition 2.3 tailored to the probabilistic setting of this chapter.

Proposition 3.2 *Let Assumptions 2.1, 3.1 and 3.3 hold. Given a realization of Algorithm 3.1, for any $\eta > 0$, if $\|\nabla f(x_k)\| > \eta$, $\alpha_k < \frac{\eta\kappa}{\nu_g}$ and D_k is κ-descent at a successful iteration k, then*

$$f(x_{k+1}) \;\leq\; f(x_k) - \frac{\kappa}{2}\,\alpha_k\,\|\nabla f(x_k)\|. \tag{3.9}$$

We thus need for (3.9) to be satisfied sufficiently often so that assuming that the gradient does not go to zero will yield a contradiction. To show that it is indeed the case, we will again rely on an argument from probability theory, that we state below and prove in Appendix A.

Lemma 3.5 *Let $\{Y_k\}$ be a sequence of nonnegative uniformly random variables, and $\{\Gamma_k\}$ a sequence of Bernoulli random variables taking the values $\ln\gamma$ and $\ln\theta$, such that*

$$\mathbb{P}\left[\Gamma_k = \ln\gamma \mid \sigma(\Gamma_0,\ldots,\Gamma_{k-1}), \sigma(Y_0,\ldots,Y_k)\right] \;\geq\; p_0.$$

We define P as the set of indexes k such that $\Gamma_k = \ln\gamma$ and $\mathcal{N} = \mathbb{N}\setminus P$. Then

$$\mathbb{P}\left[\left\{\sum_{i\in P} Y_i < \infty\right\} \bigcap \left\{\sum_{i\in\mathcal{N}} Y_i = \infty\right\}\right] \;=\; 0.$$

This property will be invoked in the proof of Lemma 3.6, which ensures that on iterations for which the gradient norm is not too small, the growth of the associated step size sequence is not too fast.

Lemma 3.6 *Let the assumptions of Theorem 3.1 hold, and assume further that Assumption 3.3 is satisfied. Let $\epsilon > 0$, and $\{K_i\}$ be the random sequence of indexes for which $\|\nabla f(X_{K_i})\| > \epsilon$. Then*

$$\mathbb{P}\left(\sum_i \mathcal{A}_{K_i} < \infty\right) \;=\; 1. \tag{3.10}$$

Proof. Consider a realization of our algorithm, and $\{k_i\}$ the corresponding realization for the random sequence $\{K_i\}$. We divide the sequence $\{K_i\}$ into $\{P_i\}$, the subsequence of $\{K_i\}$ for which the κ-descent property is verified, and $\{N_i\}$, that is the remaining subsequence. These subsequences are originally random: their realizations will be denoted by $\{p_i\}$ and $\{n_i\}$, respectively.

We begin by showing that the series $\sum_{p_i} \alpha_{p_i}$ converges. If the sequence $\{p_i\}$ is finite, such a result is immediate. Otherwise, by Lemma 3.1, we know that there exists a index j such that

$$\alpha_k < \min\left\{\varphi(\kappa\,\epsilon), \frac{\kappa\,\epsilon}{2\nu}\right\}$$

if $k \geq j$. The first term in the minimum implies that for each index $p_i \geq j$, the assumptions of Lemma 3.3 (D_{p_i} is κ-descent and $\alpha_{p_i} < \varphi(\kappa\,\epsilon) \leq \varphi(\kappa\,\|\nabla f(x_{p_i})\|)$) will be satisfied at the corresponding iteration. Consequently, every iteration of this type is successful. Besides, thanks to the second term in the minimum, we also are in the assumptions of Proposition 3.2. Thus, we have:

$$\forall\, i,\; p_i \geq j, \qquad f(x_{p_i}) - f(x_{p_i+1}) \geq \frac{\kappa}{2}\alpha_{p_i}\,\|\nabla f(x_{p_i})\| \geq \frac{\kappa\,\epsilon}{2}\alpha_{p_i}.$$

By summing on all iterations similarly to Lemma 3.1, we obtain that

$$\sum_{i,p_i\geq j} \alpha_{p_i} \leq \frac{2(f_0 - f_{\text{low}})}{\kappa\,\epsilon} < \infty \qquad \Rightarrow \qquad \sum_i \alpha_{p_i} < \infty.$$

This implies that the event $\{\sum_i A_{P_i} < \infty\}$ has probability 1.
We conclude by applying Lemma 3.5 with $\{A_k\}$ as $\{Y_k\}$, $\{\ln\gamma\, Z_k + (1 - Z_k)\ln\theta\}$ as $\{\Gamma_k\}$, with $\mathcal{P} = \{P_i\}$ and $\mathcal{N} = \{N_i\}$. The assumptions of the lemma are satisfied, and this gives us

$$\mathbb{P}\left[\left\{\sum_{P_i} A_{P_i} < \infty\right\} \bigcap \left\{\sum_{N_i} A_{N_i} = \infty\right\}\right] = 0,$$

hence the result. □

Theorem 3.2 *Consider the sequence $\{X_k\}$ of the iterates of Algorithm 3.1, under the same assumptions as Theorem 3.1. Suppose that Assumption 3.3 also holds; then*

$$\mathbb{P}\left[\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right] = 1. \tag{3.11}$$

Proof. The proof follows the one proposed in [14, Theorem 4.3] for trust-region methods based on probabilistic models. We suppose that the result does not hold, which by definition means that

$$\mathbb{P}\left(\exists\, \epsilon > 0, \exists K \subset \mathbb{N}, |K| = \infty, \forall\, k \in K, \|\nabla f(X_k)\| > \gamma\, \epsilon\right) > 0. \tag{3.12}$$

We thus also have

$$\begin{aligned} \mathbb{P}\big(\exists\, \epsilon > 0, \exists K \subset \{K_i\} \subset \mathbb{N}, |K| = \infty, \forall\, k \in K, \\ \|\nabla f(X_k)\| > \gamma\, \epsilon \;\&\; \forall\, i, \|\nabla f(X_{K_i})\| > \epsilon\big) \;>\; 0 \end{aligned} \tag{3.13}$$

where $\{K_i\}$ is the random sequence we considered in Lemma 3.6.

Considering a pair of random indexes (W, W') that satisfies the following relations:

- $0 < W < W'$;
- $\|\nabla f(X_{W'})\| \leq \epsilon$, $\|\nabla f(X_{W'+1})\| > \epsilon$, $\|\nabla f(X_{W''})\| > \gamma\, \epsilon$;
- $\forall\, W \in (W', W'')$, $\epsilon < \|\nabla f(X_W)\| \leq \gamma\, \epsilon$.

Because we are in the assumptions of Theorem 3.2, we know that there are infinitely many such intervals of this type when the event described by (3.13) happens, in which case $\{W'+1, \ldots, W''\} \subset \{K_i\}$ and we define $(W_l^{'}, W_l^{''})_l$ as the sequence of such couples.

We thus place ourselves in a realization where the event described in (3.13) happens and denote by $\{k_i\}$ and $\{(w_l^{'}, w_l^{''})\}_l$ the realizations of $\{K_i\}$ and $\left\{(W_l^{'}, W_l^{''})\right\}_l$, respectively. For all l, one has:

$$\begin{aligned} \epsilon < \left|\|\nabla f(x_{w_l^{'}})\| - \|\nabla f(x_{w_l^{''}})\|\right| &\leq \sum_{j=w_l^{'}}^{w_l^{''}-1} \left|\|\nabla f(x_j)\| - \|\nabla f(x_{j+1})\|\right| \\ &\leq \nu_g \sum_{j=w_l^{'}}^{w_l^{''}-1} \|x_j - x_{j+1}\| \\ &\leq \nu_g \left(\alpha_{w_l^{'}} + \sum_{j=w_l^{'}}^{w_l^{''}-1} \alpha_j\right). \end{aligned}$$

As $\{\alpha_k\}$ goes to 0, for l large enough, one has $\alpha_{w_l^{'}} < \frac{\epsilon}{2\,\nu_g}$, hence

$$\sum_{j=w_l^{'}+1}^{w_l^{''}-1} \alpha_j > \frac{\epsilon}{2} > 0 \;\Rightarrow \sum_{j\in\{k_i\}} \alpha_j = \infty.$$

Without loss of generality, we can assume that ϵ is the inverse of some natural number n_ϵ. If we put the previous result in probability, it gives us:

$$\mathbb{P}\left(\exists\, n_\epsilon \in \mathbb{N}, \sum_{j\in\{K_i\}} A_j = \infty\right) \;>\; 0 \tag{3.14}$$

where $\{K_i\}$ is defined according to n_ϵ. Now, from Lemma 3.6, we know that for any $n_\epsilon \in \mathbb{N}$,

$$\mathbb{P}\left(\sum_{j\in\{K_i\}} A_j = \infty\right) \;=\; 0,$$

where $\{K_i\}$ is defined the same way as in (3.14). Since the probability of a countable union of events with probability zero is still an event of probability zero, we obtain

$$\mathbb{P}\left(\bigcup_{n_\epsilon\in\mathbb{N}} \left\{\sum_{j\in\{K_i\}} A_j = \infty\right\}\right) \;=\; 0, \tag{3.15}$$

where $\{K_i\}$ is the same random sequence than in (3.14), defined according to n_ϵ. We are thus faced with contradictory statements, from which we draw the conclusion that $\lim_k \|\nabla f(X_k)\| = 0$ with probability 1. □

Like in deterministic direct search, we will rather rely on opportunistic polling than on complete polling, thus the above results will not hold. For this reason, we will develop the complexity analysis in the next section based on the convergence result of Theorem 2.1.

3.4 Complexity results

After establishing that Algorithm 3.1 shares (almost) the same convergence properties than its deterministic counterpart, we would like to provide a worst-case estimate of its performance in terms of consumption of function evaluations. Given a tolerance $\epsilon > 0$, we aim at bounding the number of iterations (and calls to the objective) necessary to satisfy

$$\inf_{0\le l\le k} \|\nabla f(X_l)\| < \epsilon.$$

Since the infimum is a random quantity, and convergence is guaranteed with probability one, a complexity result will likely hold with a given probability: this is the first type of result we will establish. Several complexity results have also been proposed that make use of the expected value on some criterion [29, 99] or derive worst-case estimates tailored to a minimum probability of satisfaction [104]. We will provide those types of results as well.

Ideally, a complexity analysis of Algorithm 3.1 would follow that of Vicente [114] for the deterministic case. However, such a study relies on guarantees provided at the iteration level through the use of PSSs or κ-descent sets. In our probabilistic environment,

it is possible that these guarantees are not met at a given iteration, thereby preventing the study of the corresponding gradient. Indeed, in this case, we may encounter an unsuccessful polling step due to the poor quality of the polling set rather than to a step size that is not small enough.

For measuring the global rate of decay of the gradient, we consider the gradient $\tilde{g}_k$ with minimum norm among $g_0, g_1, \ldots, g_k$, and denote by $\tilde{G}_k$ the corresponding random variable. Given $\epsilon > 0$, we define k_ϵ as the smallest integer k such that $\|g_k\| \leq \epsilon$, and denote the corresponding random variable by K_ϵ. We are thus interested in providing lower bounds on the probabilities $\mathbb{P}(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k}))$ (global rate) and $\mathbb{P}(K_\epsilon \leq \mathcal{O}(\epsilon^{-2}))$ (worst-case complexity bound), as close to one as possible.

We carry out the proofs in two steps. First, we identify the properties of Algorithm 3.1 that relate the number of iterations for which the polling sets are of good quality. Then, we use a probabilistic tool in conjunction with the (p, κ)-descent property to obtain our desired bounds.

3.4.1 Measuring the gradient evolution for a direct-search method

We start by looking at a realization of Algorithm 3.1. Without any assumption on the quality of the polling sets, it is possible to consider the variables z_k as realizations of the random Bernoulli variables Z_k, defined at every iteration k by

$$z_k \;=\; \begin{cases} 1 & \text{if } \operatorname{cm}(D_k, -g_k) \geq \kappa, \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

These variables will measure the quality of our polling sets. In a deterministic setting, it is possible to define a polling sequence such that there exists a κ for which every z_k is equal to 1. In a probabilistic setting, however, we would like to bound the number of z_k that are equal to 1, since it appears strongly connected to the ability of the method to produce decrease in the function value and, eventually, in the gradient norm.

Assumption 3.4 *There exist two constants $(\bar{\theta}, \bar{\gamma})$ satisfying $0 < \bar{\theta} < 1 \leq \bar{\gamma}$ such that for any $\alpha > 0$,*

$$\rho(\theta\,\alpha) \;\leq\; \bar{\theta}\,\rho(\alpha), \quad \rho(\gamma\,\alpha) \;\leq\; \bar{\gamma}\,\rho(\alpha).$$

Note that Assumption 3.4 is satisfied by classical choices of forcing function such as the monomials $\rho(\alpha) = c\,\alpha^q$, with $c > 0$ and $q > 1$. Thanks to this assumption, we are able to provide an extension of Lemma 3.1 through the result below.

Lemma 3.7 *Let Assumption 2.2, 2.4 and 3.4 hold. Then, independently of the choice of the polling sets, the step size sequence produced by a realization of Algorithm 3.1 satisfies*

$$\sum_{i=0}^{\infty} \rho(\alpha_k) \;\leq\; \beta \;=\; \frac{\bar{\gamma}}{1-\bar{\theta}} \left[\rho\left(\gamma^{-1}\,\alpha_0\right) + (f_0 - f_{\text{low}})\right]. \tag{3.17}$$

Proof. Consider a realization of Algorithm 3.1. We focus on the case of infinitely many successful iterations as the reasoning easily follows in the other case.

Let k_i be the index of the i-th successful iteration ($i \geq 1$). Define $k_0 = -1$ and $\alpha_{-1} = \gamma^{-1}\alpha_0$ for convenience. Let us rewrite $\sum_{k=0}^{\infty} \rho(\alpha_k)$ as

$$\sum_{k=0}^{\infty} \rho(\alpha_k) \;=\; \sum_{i=0}^{\infty} \sum_{k=k_i+1}^{k_{i+1}} \rho(\alpha_k), \tag{3.18}$$

and study first $\sum_{k=k_i+1}^{k_{i+1}} \rho(\alpha_k)$. According to Algorithm 3.1 and the definition of k_i, it holds

$$\begin{cases} \alpha_{k+1} \;\leq\; \gamma\alpha_k, & k = k_i, \\ \alpha_{k+1} \;=\; \theta\alpha_k, & k = k_i + 1, \ldots, k_{i+1} - 1, \end{cases}$$

which gives

$$\alpha_k \;\leq\; \gamma\theta^{k-k_i-1}\alpha_{k_i}, \quad k = k_i + 1, \ldots, k_{i+1}.$$

Hence, by the monotonicity of ρ and Assumption 3.4, we have

$$\rho(\alpha_k) \;\leq\; \bar{\gamma}\bar{\theta}^{k-k_i-1}\rho(\alpha_{k_i}), \quad k = k_i + 1, \ldots, k_{i+1}.$$

Thus

$$\sum_{k=k_i+1}^{k_{i+1}} \rho(\alpha_k) \;\leq\; \frac{\bar{\gamma}}{1-\bar{\theta}}\rho(\alpha_{k_i}). \tag{3.19}$$

Inequalities (3.18) and (3.19) imply

$$\sum_{k=0}^{\infty} \rho(\alpha_k) \;\leq\; \frac{\bar{\gamma}}{1-\bar{\theta}} \sum_{i=0}^{\infty} \rho(\alpha_{k_i}). \tag{3.20}$$

Inequality (3.20) is sufficient to conclude the proof because

$$\alpha_{k_0} \;=\; \gamma^{-1}\alpha_0 \quad \text{and} \quad \sum_{i=1}^{\infty} \rho(\alpha_{k_i}) \;\leq\; f(x_0) - f_{\text{low}},$$

according to the definition of k_i. □

Using the above bound, we can establish an upper bound on the sum $\sum_{l=0}^{k-1} z_l$, that is, on the number of times the quality of the polling sets is in our favor.

Lemma 3.8 *Given a realization of Algorithm 3.1 and a positive integer k,*

$$\sum_{l=0}^{k-1} z_l \;\leq\; \frac{\beta}{\rho\left(\min\left\{\gamma^{-1}\alpha_0, \varphi\left(\kappa \left\|\tilde{g}_k\right\|\right)\right\}\right)} + p_0\, k.$$

Proof. Consider a realization of Algorithm 3.1. For each $l \in \{0, 1, \ldots, k-1\}$, define

$$v_l \;=\; \begin{cases} 1 & \text{if } \alpha_l \;<\; \min\left\{\gamma^{-1}\alpha_0, \varphi(\kappa\|\tilde{g}_k\|)\right\}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.21}$$

A key observation for proving the lemma is

$$z_l \;\leq\; (1 - v_l) + v_l\, y_l. \tag{3.22}$$

When $v_l = 0$, inequality (3.22) is trivial; when $v_l = 1$, Lemma 3.3 implies that $y_l \geq z_l$ (since $\|\tilde{g}_k\| \leq \|\tilde{g}_{k-1}\| \leq \|\tilde{g}_l\|$), and hence inequality (3.22) holds. It suffices then to separately prove

$$\sum_{l=0}^{k-1}(1 - v_l) \;\leq\; \frac{\beta}{\rho(\min\{\gamma^{-1}\alpha_0, \varphi(\kappa\|\tilde{g}_k\|)\})} \tag{3.23}$$

and

$$\sum_{l=0}^{k-1} v_l\, y_l \;\leq\; p_0\, k. \tag{3.24}$$

Because of Lemma 3.7, inequality (3.23) is justified by the fact that

$$1 - v_l \;\leq\; \frac{\rho(\alpha_l)}{\rho(\min\{\gamma^{-1}\alpha_0, \varphi(\kappa\|\tilde{g}_k\|)\})},$$

which in turn is guaranteed by the definition (3.21) and the monotonicity of ρ.

Now consider inequality (3.24). If $v_l = 0$ for all $l \in \{0, 1, \ldots, k-1\}$, then (3.24) holds. Consider thus that $v_l = 1$ for some $l \in \{0, 1, \ldots, k-1\}$. Let $\bar{l}$ be the largest one of such integers. Then

$$\sum_{l=0}^{k-1} v_l\, y_l \;=\; \sum_{l=0}^{\bar{l}} v_l\, y_l, \tag{3.25}$$

and we can estimate the sum on the right-hand side. For each $l \in \left\{0, 1, \ldots, \bar{l}\right\}$, Algorithm 3.1 together with the definitions of v_l and y_l gives

$$\begin{cases} \alpha_{l+1} \;=\; \min\{\gamma\,\alpha_l, \alpha_{\max}\} \;=\; \gamma\alpha_l & \text{if } v_l\, y_l = 1, \\ \alpha_{l+1} \;\geq\; \theta\alpha_l & \text{if } v_l\, y_l = 0, \end{cases}$$

which implies

$$\alpha_{\bar{l}+1} \;\geq\; \alpha_0 \prod_{l=0}^{\bar{l}} \left(\gamma^{v_l y_l}\theta^{1 - v_l y_l}\right). \tag{3.26}$$

On the other hand, since $v_{\bar{l}} = 1$, we have $\alpha_{\bar{l}} \leq \gamma^{-1}\alpha_0$, hence $\alpha_{\bar{l}+1} \leq \alpha_0$. Consequently, by taking logarithms, one can obtain from inequality (3.26) that

$$0 \;\geq\; \ln(\gamma\,\theta^{-1}) \sum_{l=0}^{\bar{l}} v_l\, y_l + (\bar{l} + 1)\ln\theta,$$

which leads to

$$\sum_{l=0}^{\bar{l}} v_l\, y_l \;\leq\; \frac{\ln\theta}{\ln(\gamma^{-1}\theta)}(\bar{l} + 1) \;=\; p_0(\bar{l} + 1) \;\leq\; p_0 k \tag{3.27}$$

since $\ln(\gamma^{-1}\theta) < 0$. Inequality (3.24) is finally obtained by combining inequalities (3.25) and (3.27). □

We have thus a direct relationship between the sum of the z_l and the minimum gradient norm at every iteration. Note that this is a deterministic result, that generalizes the reasoning made by Vicente [114] for the worst-case complexity of deterministic direct search with respect to the forcing functions. Indeed, it applies to more general choices of ρ than monomials, as long as they satisfy Assumption 3.4, however it requires to choose $\gamma > 1$.

The following proposition provides the equivalent of Theorem 2.3 in our framework. Here again, the deterministic property is obtained through a reasoning on realizations.

Proposition 3.3 *Assume that $\gamma > 1$ and consider a realization of Algorithm 3.1. If, for each $k \geq 0$,*

$$\mathrm{cm}(D_k, -g_k) \ \geq \ \kappa \tag{3.28}$$

and

$$\epsilon \ \leq \ \frac{\gamma}{\kappa\,\alpha_0}\rho(\gamma^{-1}\alpha_0) + \frac{\nu_g\,\alpha_0}{2\kappa\,\gamma}, \tag{3.29}$$

then

$$k_\epsilon \ \leq \ \frac{\beta}{(1-p_0)\rho[\varphi(\kappa\,\epsilon)]}.$$

Proof. By the definition of k_ϵ, we have $\|\tilde{g}_{k_\epsilon-1}\| \geq \epsilon$. Therefore, from Lemma 3.8 (which holds also with $\|\tilde{g}_k\|$ replaced by $\|\tilde{g}_{k-1}\|$) and the monotonicity of ρ and φ, we obtain

$$\sum_{l=0}^{k_\epsilon-1} z_l \ \leq \ \frac{\beta}{\rho\left(\min\left\{\gamma^{-1}\alpha_0, \varphi(\kappa\,\epsilon)\right\}\right)} + p_0\,k_\epsilon. \tag{3.30}$$

According to (3.28), $z_k = 1$ for each $k \geq 0$. By the definition of φ, inequality (3.29) implies

$$\varphi(\kappa\epsilon) \ \leq \ \varphi\left[\frac{\rho(\gamma^{-1}\alpha_0)}{\gamma^{-1}\alpha_0} + \frac{\nu_g}{2}\gamma^{-1}\alpha_0\right] \ \leq \ \gamma^{-1}\alpha_0. \tag{3.31}$$

Hence (3.30) reduces to

$$k_\epsilon \ \leq \ \frac{\beta}{\rho\left[\varphi(\kappa\,\epsilon)\right]} + p_0\,k_\epsilon.$$

Since $\gamma > 1$, one has, from (3.6), $p_0 < 1$, and the proof is completed. □

3.4.2 Main complexity results and comparison with the deterministic setting

We now come back to the probabilistic setting, and study the probability $\mathbb{P}(\|\tilde{G}_k\| \leq \epsilon)$ (and equivalently $\mathbb{P}(K_\epsilon \leq k)$) with the help of Lemma 3.8. First we present a universal lower bound for this probability, which holds without any assumption on the probabilistic behavior of $\{\mathfrak{D}_k\}$.

Using this bound, we prove that $\mathbb{P}(\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k}))$ and $\mathbb{P}(K_\epsilon \leq \mathcal{O}(\epsilon^{-2}))$ are overwhelmingly high when $\rho(\alpha) = c\,\alpha^2/2$ (Corollaries 3.1 and 3.2), which will be given as special cases of the results for general forcing functions (Theorems 3.3 and 3.4).

Lemma 3.9 *If*

$$\epsilon \;\leq\; \frac{\gamma}{\kappa\,\alpha_0}\rho(\gamma^{-1}\alpha_0) + \frac{\nu_g\,\alpha_0}{2\kappa\,\gamma}, \tag{3.32}$$

then

$$\mathbb{P}\left(\|\tilde{G}_k\| \leq \epsilon\right) \;\geq\; 1 - \pi_k\!\left(\frac{\beta}{k\rho[\varphi(\kappa\,\epsilon)]} + p_0\right), \tag{3.33}$$

where $\pi_k(\lambda) = \mathbb{P}\left(\sum_{l=0}^{k-1} Z_l \leq \lambda k\right)$.

Proof. According to Lemma 3.8 and the monotonicity of ρ and φ, we have

$$\left\{\|\tilde{G}_k\| \geq \epsilon\right\} \;\subset\; \left\{\sum_{l=0}^{k-1} Z_l \leq \frac{\beta}{\rho(\min\{\gamma^{-1}\alpha_0, \varphi(\kappa\,\epsilon)\})} + p_0\,k\right\}. \tag{3.34}$$

Again, as in (3.31), by the definition of φ, inequality (3.32) implies $\varphi(\kappa\,\epsilon) \leq \gamma^{-1}\alpha_0$. Thus we rewrite (3.34) as

$$\left\{\|\tilde{G}_k\| \geq \epsilon\right\} \;\subset\; \left\{\sum_{l=0}^{k-1} Z_l \leq \frac{\beta}{\rho[\varphi(\kappa\,\epsilon)]} + p_0\,k\right\},$$

which gives us inequality (3.33) according to the definition of π_k. □

This lemma enables us to lower bound $\mathbb{P}(\|\tilde{G}_k\| \leq \epsilon)$ by just focusing on the function π_k, which is a classical object in probability theory. Various lower bounds can then be established under different assumptions on $\{\mathfrak{D}_k\}$.

Given the assumption that $\{\mathfrak{D}_k\}$ is (p,κ)-descent, Definition 3.2 implies that

$$\mathbb{P}(Z_0 = 1) \;\geq\; p \quad \text{and} \quad \mathbb{P}\left(Z_k = 1 \mid Z_0, \ldots, Z_{k-1}\right) \;\geq\; p \quad (k \geq 1). \tag{3.35}$$

It is known that the lower tail of $\sum_{l=0}^{k-1} Z_l$ obeys a Chernoff type bound, even when conditioning to the past replaces the more traditional assumption of independence of the Z_k's (see, for instance, [52, Problem 1.7] and [49, Lemma 1.18]). We present such a bound in Lemma 3.10 below and reserve its proof for Appendix A.

Lemma 3.10 *Suppose that* $\{\mathfrak{D}_k\}$ *is* (p,κ)*-descent and* $\lambda \in (0,p)$*. Then*

$$\pi_k(\lambda) \;\leq\; \exp\left[-\frac{(p-\lambda)^2}{2p}k\right]. \tag{3.36}$$

We can now present the main results of this section. To this end, we will assume that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$, which cannot be fulfilled unless $\gamma > 1$ (since $p_0 = 1$ if $\gamma = 1$).

Theorem 3.3 *Suppose that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$ and that*

$$k \;\geq\; \frac{(1+\delta)\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}, \tag{3.37}$$

for some positive number δ, and where ϵ satisfies

$$\epsilon \;\leq\; \frac{\gamma}{\kappa\,\alpha_0}\rho(\gamma^{-1}\alpha_0) + \frac{\nu_g\,\alpha_0}{2\kappa\,\gamma}. \tag{3.38}$$

Then

$$\mathbb{P}\left(\|\tilde{G}_k\| \leq \epsilon\right) \;\geq\; 1 - \exp\left[-\frac{(p-p_0)^2\delta^2}{2p(1+\delta)^2}k\right]. \tag{3.39}$$

Proof. According to Lemma 3.9 and the monotonicity of π_k,

$$\mathbb{P}\left(\|\tilde{G}_k\| \leq \epsilon\right) \;\geq\; 1 - \pi_k\left(\frac{\beta}{k\rho[\varphi(\kappa\,\epsilon)]} + p_0\right) \;\geq\; 1 - \pi_k\left(\frac{p-p_0}{1+\delta} + p_0\right).$$

Then inequality (3.39) follows directly from Lemma 3.10. □

Theorem 3.3 reveals that, when ϵ is an arbitrary number but small enough to verify (3.38) and the iteration counter satisfies (3.37) for some positive number δ, the norm of the gradient is below ϵ with overwhelmingly high probability. Note that ϵ is related to k through (3.37). In fact, if $\rho\circ\varphi$ is invertible (which is true when the forcing function is a multiple of α^q with $q > 1$), then, for any positive integer k, sufficiently large to satisfy (3.37) and (3.38) all together, one can set

$$\epsilon \;=\; \frac{1}{\kappa}(\rho\circ\varphi)^{-1}\left(\frac{(1+\delta)\beta}{p-p_0}\frac{1}{k}\right), \tag{3.40}$$

and what we have in (3.39) can then be read as (since ϵ and k satisfy the assumptions of Theorem 3.3)

$$\mathbb{P}\left(\|\tilde{G}_k\| \leq \kappa^{-1}(\rho\circ\varphi)^{-1}(\mathcal{O}(1/k))\right) \;\geq\; 1 - \exp(-C\,k),$$

with C a positive constant.

A particular case is when the forcing function is a multiple of the square of the step size

$$\rho(\alpha) \;=\; \frac{1}{2}c\,\alpha^2,$$

where it is easy to check that

$$\varphi(t) \;=\; \frac{2\,t}{c+\nu_g}.$$

One can then obtain the global rate $\|\tilde{G}_k\| \leq \mathcal{O}(1/\sqrt{k})$ with overwhelmingly high probability, matching [114] for deterministic direct search, as it is shown below in Corollary 3.1, which is just an application of Theorem 3.3 for this particular forcing function. For simplicity, we will set $\delta = 1$.

Corollary 3.1 *Suppose that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$, $\rho(\alpha) = c\,\alpha^2/2$, and*

$$k \;\geq\; \frac{4\gamma^2\beta}{c(p-p_0)\,\alpha_0^2}. \tag{3.41}$$

Then

$$\mathbb{P}\left(\|\tilde{G}_k\| \leq \left(\frac{\beta^{\frac{1}{2}}(c+\nu_g)}{c^{\frac{1}{2}}(p-p_0)^{\frac{1}{2}}\kappa}\right)\frac{1}{\sqrt{k}}\right) \;\geq\; 1-\exp\left[-\frac{(p-p_0)^2}{8p}k\right]. \tag{3.42}$$

Proof. As in (3.40), with $\delta = 1$, let

$$\epsilon \;=\; \frac{1}{\kappa}(\rho\circ\varphi)^{-1}\left(\frac{2\beta}{p-p_0}\frac{1}{k}\right). \tag{3.43}$$

Then, by straightforward calculations, we have

$$\epsilon \;=\; \left(\frac{\beta^{\frac{1}{2}}(c+\nu_g)}{c^{\frac{1}{2}}(p-p_0)^{\frac{1}{2}}\kappa}\right)\frac{1}{\sqrt{k}}. \tag{3.44}$$

Moreover, inequality (3.41) gives us

$$\epsilon \;\leq\; \frac{\beta^{\frac{1}{2}}(c+\nu_g)}{c^{\frac{1}{2}}(p-p_0)^{\frac{1}{2}}\kappa}\left(\frac{4\gamma^2\beta}{c(p-p_0)\alpha_0^2}\right)^{-\frac{1}{2}} \;=\; \frac{(c+\nu_g)\,\alpha_0}{2\kappa\,\gamma}. \tag{3.45}$$

Definition (3.43) and inequality (3.45) guarantee that k and ϵ satisfy (3.37) and (3.38) for $\rho(\alpha) = c\,\alpha^2/2$ and $\delta = 1$. Hence we can plug (3.44) into (3.39), which finally yields (3.42). □

Based on Lemmas 3.9 and 3.10 (or directly on Theorem 3.3), one can lower bound $\mathbb{P}(K_\epsilon \leq k)$ and arrive at a worst-case complexity result.

Theorem 3.4 *Suppose that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$ and*

$$\epsilon \;\leq\; \frac{\gamma}{\kappa\,\alpha_0}\rho(\gamma^{-1}\alpha_0) + \frac{\nu_g\,\alpha_0}{2\kappa\,\gamma}.$$

Then, for each $\delta > 0$,

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil\frac{(1+\delta)\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}\right\rceil\right) \;\geq\; 1-\exp\left[-\frac{\beta(p-p_0)\delta^2}{2p(1+\delta)\rho[\varphi(\kappa\,\epsilon)]}\right]. \tag{3.46}$$

Proof. Letting

$$k \;=\; \left\lceil\frac{(1+\delta)\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}\right\rceil$$

we have $\mathbb{P}(K_\epsilon \leq k) = \mathbb{P}(\|\tilde{G}_k\| \leq \epsilon)$ and then inequality (3.46) follows from Theorem 3.3 as

$$k \;\geq\; \frac{(1+\delta)\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}.$$

□

Since $\rho(\alpha) = o(\alpha)$ (from the definition of the forcing function ρ) and $\varphi(\kappa\,\epsilon) \leq 2\nu_g^{-1}\kappa\,\epsilon$ (from the definition of φ), it holds that the lower bound in (3.46) goes to one faster than $1 - \exp(-\epsilon^{-1})$. Hence, we conclude from Theorem 3.4 that direct search based on probabilistic descent exhibits a worst-case complexity bound in number of iterations of the order of $1/\rho[\varphi(\kappa\,\epsilon)]$ with overwhelmingly high probability, matching Proposition 3.3 for deterministic direct search. To see this effect more clearly, we present in Corollary 3.2 a specification of Theorem 3.4 when $\rho(\alpha) = c\,\alpha^2/2$, taking $\delta = 1$ as in Corollary 3.1. The worst-case complexity bound is then of the order of $1/\epsilon^2$ with overwhelmingly high probability, matching [114] for deterministic direct search. The same matching happens when $\rho(\alpha)$ is a power of α with exponent q, where the bound is $\mathcal{O}(\epsilon^{-\frac{q}{\min\{q-1,1\}}})$.

Corollary 3.2 *Suppose that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$, $\rho(\alpha) = c\,\alpha^2/2$, and*

$$\epsilon \;\leq\; \frac{(c+\nu_g)\alpha_0}{2\kappa\,\gamma}.$$

Then

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{\beta\,(c+\nu_g)^2}{c(p-p_0)\kappa^2}\epsilon^{-2}\right\rceil\right) \;\geq\; 1 - \exp\left[-\frac{\beta(p-p_0)(c+\nu_g)^2}{8c\,p\,\kappa^2}\epsilon^{-2}\right]. \tag{3.47}$$

It is important to understand how the worst-case complexity bound (3.47) depends on the dimension n of the problem. For this purpose, we first need to make explicit the dependence on n of the constant in $K_\epsilon \leq \lceil \beta\,(c+\nu_g)^2/[c(p-p_0)\,\kappa^2]\epsilon^{-2}\rceil$, which can only come from p and κ and is related to the choice of $\mathfrak{D}_k$.

One can choose $\mathfrak{D}_k$ as m directions uniformly independently distributed on the unit sphere, with m independent of n, in which case p is a constant larger than p_0 and $\kappa = \tau/\sqrt{n}$ for some constant $\tau > 0$ (both p and τ are totally determined by γ and θ without dependence on m or n; see Section 3.5). In such a case, from Corollary 3.2,

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{\beta\,(c+\nu_g)^2}{c\,(p-p_0)\,\tau^2}\left(n\,\epsilon^{-2}\right)\right\rceil\right) \;\geq\; 1 - \exp\left[-\frac{\beta\,(p-p_0)(c+\nu_g)^2}{8\,c\,p\,\kappa^2}\epsilon^{-2}\right].$$

To derive a worst-case complexity bound in terms of the number of function evaluations, one just needs then to see that each iteration of Algorithm 3.1 costs at most m function evaluations. Thus, if K_ϵ^f represents the number of function evaluations within K_ϵ iterations, we obtain

$$\mathbb{P}\left(K_\epsilon^f \leq \left\lceil \frac{\beta\,(c+\nu_g)^2}{c\,(p-p_0)\,\tau^2}\left(n\,\epsilon^{-2}\right)\right\rceil m\right) \;\geq\; 1 - \exp\left[-\frac{\beta(p-p_0)(c+\nu_g)^2}{8\,c\,p\,\tau^2}n\,\epsilon^{-2}\right]. \tag{3.48}$$

We observe from the expression of β determined in Lemma 3.7 that its value does not depend on the problem dimension, nor on the number of evaluations. As mentioned above, p, τ and p_0 do not depend on n nor m. In our general setting, there is also no

reason to assume that the Lipschitz constant depends on the dimension, while c is chosen independently of it. As a result, the worst-case complexity bound is then $\mathcal{O}\left(m\, n\, \epsilon^{-2}\right)$ with overwhelmingly high probability, which is clearly better than the corresponding bound $\mathcal{O}(n^2\epsilon^{-2})$ for deterministic direct search presented in Section 2.3 if m is chosen an order of magnitude smaller than n.

3.4.3 Additional complexity properties

High probability iteration complexity Given a confidence level P, the following theorem presents an explicit bound for the number of iterations which can guarantee that $\|\tilde{G}_k\| \leq \epsilon$ holds with probability at least P. Bounds of this type are interesting in practice and have been considered in the theoretical analysis of randomized algorithms (see, for instance, [100, 104]).

Theorem 3.5 *Suppose that $\{\mathfrak{D}_k\}$ is (p, κ)-descent with $p > p_0$. Then for any*

$$\epsilon \;\leq\; \frac{\gamma}{\kappa\,\alpha_0}\rho(\gamma^{-1}\alpha_0) + \frac{\nu_g\,\alpha_0}{2\kappa\,\gamma}$$

and $P \in (0,1)$, it holds $\mathbb{P}(\|\tilde{G}_k\| \leq \epsilon) \geq P$ whenever

$$k \;\geq\; \frac{3\beta}{2(p-p_0)\rho[\varphi(\kappa\,\epsilon)]} - \frac{3\,p\,\ln(1-P)}{(p-p_0)^2}. \tag{3.49}$$

Proof. By Theorem 3.3, $\mathbb{P}(\|\tilde{G}_k\| \leq \epsilon) \geq P$ is achieved when

$$k \;\geq\; \max\left\{\frac{(1+\delta)\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}, -\frac{2p(1+\delta)^2\ln(1-P)}{\delta^2(p-p_0)^2}\right\} \tag{3.50}$$

for some positive number δ. Hence it suffices to show that the right-hand side of (3.49) is bigger than that of (3.50) for properly chosen δ. For simplicity, denote

$$c_1 \;=\; \frac{\beta}{(p-p_0)\rho[\varphi(\kappa\,\epsilon)]}, \quad c_2 \;=\; -\frac{2p\,\ln(1-P)}{(p-p_0)^2}.$$

Let us consider the positive number δ such that

$$(1+\delta)c_1 \;=\; \frac{(1+\delta)^2}{\delta^2}c_2.$$

It is easy to check that

$$\delta \;=\; \frac{1}{2c_1}\left(c_2 + \sqrt{c_2^2 + 4c_1c_2}\right) \;\leq\; \frac{1}{2} + \frac{3c_2}{2c_1}.$$

Thus

$$\max\left\{(1+\delta)c_1, \frac{(1+\delta)^2}{\delta^2}c_2\right\} \;=\; (1+\delta)c_1 \;\leq\; \frac{3}{2}(c_1+c_2),$$

which completes the proof. □

Expected minimum norm gradient We now study how $\mathbb{E}\left[\|\tilde{G}_k\|\right]$ behaves relatively to the iteration counter k. For simplicity, we consider the special case of the forcing function $\rho(\alpha) = c\,\alpha^2/2$; we then have the following result.

Proposition 3.4 *Suppose that $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$, $\rho(\alpha) = c\,\alpha^2/2$, and*

$$k \;\geq\; \frac{4\gamma^2\beta}{c(p-p_0)\alpha_0^2}.$$

Then

$$\mathbb{E}\left[\|\tilde{G}_k\|\right] \;\leq\; c_3 k^{-\frac{1}{2}} + \|g_0\| \exp\left(-c_4\, k\right),$$

where

$$c_3 \;=\; \frac{\beta^{\frac{1}{2}}(c+\nu_g)}{c^{\frac{1}{2}}(p-p_0)^{\frac{1}{2}}\kappa}, \quad c_4 \;=\; \frac{(p-p_0)^2}{8\,p}.$$

Proof. Let us define a random variable H_k as

$$H_k \;=\; \begin{cases} c_3\, k^{-\frac{1}{2}} & \text{if } \|\tilde{G}_k\| \leq c_3\, k^{-\frac{1}{2}}, \\ \|g_0\| & \text{otherwise.} \end{cases}$$

Then $\|\tilde{G}_k\| \leq H_k$, and hence

$$\mathbb{E}\left[\|\tilde{G}_k\|\right] \;\leq\; \mathbb{E}\left[H_k\right] \;\leq\; c_3\, k^{-\frac{1}{2}} + \|g_0\| \mathbb{P}\left(\|\tilde{G}_k\| > c_3\, k^{-\frac{1}{2}}\right).$$

Therefore it suffices to notice

$$\mathbb{P}\left(\|\tilde{G}_k\| > c_3\, k^{-\frac{1}{2}}\right) \;\leq\; \exp(-c_4\, k),$$

which is a straightforward application of Corollary 3.1. □

Recall that in deterministic direct search employing a forcing function $\rho(\alpha) = c\,\alpha^2/2$, $\|\tilde{g}_k\|$ decays with $\mathcal{O}(k^{-\frac{1}{2}})$ when k tends to infinity. Proposition 3.4 shows that $\mathbb{E}\left[\|\tilde{G}_k\|\right]$ behaves in a similar way in direct search based on probabilistic descent, in that its decaying rate gets asymptotically close to the deterministic one.

Expected number of iterations and function evaluations Complementary to the study of the expected gradient norm, we analyze the expected behavior of K_ϵ. Cartis and Scheinberg [29] provide such results for line-search and adaptive cubic regularization frameworks based on probabilistic models.

Proposition 3.5 *Suppose $\{\mathfrak{D}_k\}$ is (p,κ)-descent with $p > p_0$, $\rho(\alpha) = c\,\alpha^2/2$, and*

$$\epsilon \;\leq\; \frac{(c+\nu_g)\alpha_0}{2\,\kappa\,\gamma}.$$

Then

$$\mathbb{E}\left[K_\epsilon\right] \;\leq\; c_5 \epsilon^{-2} + \frac{1}{1-\exp(-c_4)},$$

where

$$c_5 \;=\; c_3^2 \;=\; \frac{\beta\,(c+\nu_g)}{c\,(p-p_0)\,\kappa^2},$$

and c_3, c_4 are defined as in Proposition 3.4.

Proof. Since K_ϵ takes integer values, one has:

$$\begin{aligned}
\mathbb{E}\left[K_\epsilon\right] &= \sum_{k<c_5\,\epsilon^{-2}} \mathbb{P}\left(K_\epsilon > k\right) + \sum_{k\geq c_5\,\epsilon^{-2}} \mathbb{P}\left(K_\epsilon > k\right) \\
&\leq c_5\,\epsilon^{-2} + \sum_{k\geq c_5\,\epsilon^{-2}} \mathbb{P}\left(K_\epsilon > k\right) \\
&\leq c_5\,\epsilon^{-2} + \sum_{k\geq c_5\,\epsilon^{-2}} \mathbb{P}\left(\|\tilde{G}_k\| > \epsilon\right).
\end{aligned}$$

Then, for terms in the sum, we have that

$$k \geq c_5\,\epsilon^{-2} \geq \frac{\beta\,(c+\nu_g)}{c\,(p-p_0)\,\kappa^2}\,\left[\frac{(c+\nu_g)\alpha_0}{2\,\kappa\,\gamma}\right]^{-2} = \frac{4\,\beta\,\gamma^2}{c(p-p_0)\,\alpha_0^2},$$

so we are in the assumptions of Corollary 3.1 and this gives

$$\mathbb{P}\left(K_\epsilon > k\right) \;=\; \mathbb{P}\left(\|\tilde{G}_k\| > \epsilon\right) \;\leq\; \mathbb{P}\left(\|\tilde{G}_k\| > c_3\,k^{-\frac{1}{2}}\right) \;\leq\; \exp(-c_4\,k)$$

holds. It then suffices to bound the resulting geometric progression by the sum on all $k \in \mathbb{N}$ to arrive at the desired result. □

It immediately follows that the expected number of function evaluations satisfies the bound stated below:

$$\mathbb{E}\left[K_\epsilon^f\right] \;\leq\; \mathcal{O}\left(\frac{m\,\kappa^{-2}\,\epsilon^{-2}}{p-p_0}\right) + m\,\mathcal{O}(1). \tag{3.51}$$

The bound obtained matches those of Cartis and Scheinberg [29] for line-search schemes in terms of powers of ϵ; we also emphasize that both our bound and theirs exhibit a dependence on the inverse of $p - p_0$.

3.5 A practical implementation of a probabilistic descent set sequence

We describe below how the use of uniformly distributed directions on the unit sphere allows to define a probabilistically descent sequence. We start by providing a bound of the cosine measure of such a set, expressed at a given (deterministic) vector.

Lemma 3.11 *Let $\mathfrak{D}_k$ be a set of m random vectors independently and identically distributed on the unit sphere, $v \in \mathbb{R}^n$ and $\tau \in [0, \sqrt{n}]$. One has:*

$$\mathbb{P}\left(\mathrm{cm}\,(\mathfrak{D}_k, v) \;\geq\; \frac{\tau}{\sqrt{n}}\right) \geq 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\,\pi}}\right)^m. \tag{3.52}$$

Proof. From the definition of $\mathrm{cm}\,(\mathfrak{D}, v)$, the result trivially holds if $v = 0$. Therefore in the rest of the proof we assume that $\|v\| \neq 0$ (and without loss of generality, that $\|v\| = 1$). One thus has:

$$\mathbb{P}\left(\mathrm{cm}\,(\mathfrak{D}_k, v) \;\geq\; \frac{\tau}{\sqrt{n}}\right) \;=\; 1 - \left[1 - \mathbb{P}\left(\mathfrak{d}^\top v \geq \frac{\tau}{\sqrt{n}}\right)\right]^m,$$

where $\mathfrak{d}$ is a random vector uniformly distributed on the unit sphere.

Since the distribution of $\mathfrak{d}$ is uniform on the unit sphere, the probability we are interested in is proportional to the area A of the spherical cap

$$\left\{d \in \mathbb{R}^n \;: \|d\| = 1 \text{ and } d^\top v \geq \kappa\right\}$$

of unit radius and height

$$h \;=\; 1 - \kappa.$$

Recalling the area formula for spherical caps, we have

$$A \;=\; \frac{1}{2} A_n \,\mathrm{I}\!\left(2h - h^2,\, \frac{n-1}{2},\, \frac{1}{2}\right) \;=\; \frac{1}{2} A_n \,\mathrm{I}\!\left(1 - \kappa^2,\, \frac{n-1}{2},\, \frac{1}{2}\right),$$

where A_n is the area of the unit sphere in $\mathbb{R}^n$, and I denotes the regularized incomplete Beta function [1] defined by

$$\mathrm{I}\,(u, a, b) \;=\; \frac{1}{\mathrm{B}(a, b)} \int_0^u t^{a-1}(1-t)^{b-1} dt, \tag{3.53}$$

with B being the Beta function. Hence

$$\varrho(\kappa) \;=\; \frac{1}{2}\,\mathrm{I}\!\left(1 - \kappa^2,\, \frac{n-1}{2},\, \frac{1}{2}\right), \tag{3.54}$$

and our goal is to establish that

$$\varrho(\kappa) \;\geq\; \frac{1}{2} - \kappa\sqrt{\frac{n}{2\,\pi}}. \tag{3.55}$$

When $n = 2$, by plugging (3.53) into (3.54), calculating the integral, and noticing $\mathrm{B}(\frac{1}{2}, \frac{1}{2}) = \pi$, we have

$$\varrho(\kappa) \;=\; \frac{1}{2}\,\mathrm{I}\!\left(1 - \kappa^2,\, \frac{1}{2},\, \frac{1}{2}\right) \;=\; \frac{1}{\pi}\arcsin\sqrt{1 - \kappa^2} \;=\; \frac{1}{2} - \frac{1}{\pi}\arcsin\kappa \;\geq\; \frac{1}{2} - \frac{\kappa}{2},$$

and therefore inequality (3.55) is true. To prove (3.55) in the situation of $n \geq 3$, we examine (3.53) and find that

$$\begin{aligned} \mathrm{I}\,(u,a,b) &= 1 - \frac{1}{\mathrm{B}(a,b)} \int_u^1 t^{a-1}(1-t)^{b-1} dt \\ &\geq 1 - \frac{1}{\mathrm{B}(a,b)} \int_u^1 (1-t)^{b-1} dt \\ &= 1 - \frac{(1-u)^b}{b\,\mathrm{B}(a,b)} \end{aligned}$$

when $a \geq 1$. Hence, using equation (3.54), we obtain

$$\varrho(\kappa) \;\geq\; \frac{1}{2} - \frac{\kappa}{\mathrm{B}(\frac{n-1}{2}, \frac{1}{2})}$$

when $n \geq 3$. Thus we can arrive at inequality (3.55) as long as

$$\mathrm{B}\left(\frac{n-1}{2}, \frac{1}{2}\right) \;\geq\; \sqrt{\frac{2\pi}{n}}. \tag{3.56}$$

Inequality (3.56) is justified by the facts

$$\mathrm{B}\left(\frac{n-1}{2}, \frac{1}{2}\right) \;=\; \frac{\Gamma(\frac{n-1}{2})\,\Gamma(\frac{1}{2})}{\Gamma(\frac{n}{2})} \;=\; \frac{\Gamma(\frac{n-1}{2})}{\Gamma(\frac{n}{2})}\sqrt{\pi},$$

and

$$\Gamma\left(\frac{n}{2}\right) \;\leq\; \left[\Gamma\left(\frac{n-1}{2}\right)\Gamma\left(\frac{n+1}{2}\right)\right]^{\frac{1}{2}} \;=\; \sqrt{\frac{n-1}{2}}\,\Gamma\left(\frac{n-1}{2}\right),$$

the second of which coming from the fact that Γ is log-convex, meaning that $\ln\Gamma$ is convex [7, Theorem 2.1].

□

The above results are valid when the vector v is deterministic; in our algorithmic framework, we study the cosine measure expressed at random vectors with conditioning to the past. The result still holds, as proved in the following theorem.

Theorem 3.6 *Consider a set sequence $\{\mathfrak{D}_k\}_k$, where all $\mathfrak{D}_k$ are drawn independently from the same distribution, as described in Proposition 3.11. Consider that this sequence is used in Algorithm 3.1. Then, for any $\tau \in [0, \sqrt{n}]$:*

$$\mathbb{P}\left(\mathrm{cm}\,(\mathfrak{D}_k, -G_k) > \frac{\tau}{\sqrt{n}} \middle| \sigma_{k-1}\right) \;\geq\; 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\,\pi}}\right)^m. \tag{3.57}$$

where $\sigma_{k-1} = \sigma\,(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1})$ (with the convention that σ_{-1} represents conditioning on the whole set of events, i.e. the absence of conditioning).

Proof. Consider the couple of variables $(\mathfrak{D}_k, G_k)$. By definition, G_k is independent of $\mathfrak{D}_k$. Let Ψ be defined as follows:

$$\begin{array}{rccl} \Psi: & \mathbb{R}^n \times \mathbb{R}^{n\times m} & \mapsto & \{0,1\} \\ & (g, D) & \mapsto & \mathbf{1}\left(\mathrm{cm}\left(D, g\right) > \kappa\right). \end{array}$$

One has $\mathbb{E}\left[|\Psi(G_k, \mathfrak{D}_k)|\right] < \infty$. From [54, Example 5.1.5] (see also [32, page 148]), we know that:

$$\begin{array}{rcl} \mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \dfrac{\tau}{\sqrt{n}} \,\middle|\, \sigma(G_k)\right) & = & \mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \dfrac{\tau}{\sqrt{n}} \,\middle|\, G_k\right) \\ & = & \mathbb{E}\left[\Psi(G_k, D_k)\right] \;=\; h(G_k), \end{array}$$

where h is defined as follows:

$$\begin{array}{rccl} h: & \mathbb{S}^{n-1} & \mapsto & \mathbb{R} \\ & g & \mapsto & h(g) = \mathbb{E}(\Psi(g_k, \mathfrak{D}_k)). \end{array}$$

Now, using Lemma 3.11, we obtain for any unitary vector g:

$$\begin{array}{rcl} h(g) & = & 1 \times \mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -g\right) > \dfrac{\tau}{\sqrt{n}}\right) + 0 \times \mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -g\right) \leq \dfrac{\tau}{\sqrt{n}}\right) \\ & \geq & 1 - \left(\dfrac{1}{2} + \dfrac{\tau}{\sqrt{2\,\pi}}\right)^m, \end{array}$$

so h is constant on all realizations of G_k. This leads to

$$\mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \frac{\tau}{\sqrt{n}} \,\middle|\, G_k\right) \;\geq\; 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\,\pi}}\right)^m, \tag{3.58}$$

so this random variable has a constant value.

What is left to prove is that conditioning to G_k gives the same result as conditioning to σ_{k-1}. Because G_k can be seen as a function of $\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}$, one has:

$$\sigma(G_k) \;\subset\; \sigma_{k-1}.$$

Besides, as the random variable $\mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \frac{\tau}{\sqrt{n}} \middle| G_k\right) = g(G_k)$ is a constant, it is σ_{k-1}-measurable. We thus are in the assumptions of [54, Theorem 5.1.5] and we can conclude that

$$\mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \frac{\tau}{\sqrt{n}} \,\middle|\, \sigma_{k-1}\right) \;=\; \mathbb{P}\left(\mathrm{cm}\left(\mathfrak{D}_k, -G_k\right) > \frac{\tau}{\sqrt{n}} \,\middle|\, G_k\right),$$

from which the result (3.57) follows. □

A first consequence of Theorem 3.6 is that one can construct a set sequence satisfying the probabilistic descent property.

Corollary 3.3 *Let the assumptions of Theorem 3.6 hold. Given $\tau \in [0, \sqrt{n}]$, the polling sets $\{\mathfrak{D}_k\}$ form a $(p, \tau/\sqrt{n})$-descent sequence with*

$$p \;\leq\; 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\pi}}\right)^m. \tag{3.59}$$

Thanks to Theorem 3.6, we can also give a condition on the number m of search directions generated at each step to verify the (p, κ)-descent property.

Corollary 3.4 *Let the assumptions of Theorem 3.6 hold. Then, by choosing the number of directions to be*

$$m \;>\; \log_2\left(1 - \frac{\ln\theta}{\ln\gamma}\right), \tag{3.60}$$

it exists $p \geq p_0$ and $\tau > 0$ determined by θ and γ such that the polling set sequence is $(p, \tau/\sqrt{n})$-descent.

Proof. Let m_0 be the minimal integer that satisfies

$$m_0 \;>\; \log_2\left(1 - \frac{\ln\theta}{\ln\gamma}\right). \tag{3.61}$$

Then $m \geq m_0$. Given inequality (3.61), we have

$$1 - \left(\frac{1}{2}\right)^{m_0} \;>\; 1 - \left(1 - \frac{\ln\theta}{\ln\gamma}\right)^{-1} \;=\; p_0.$$

Thus, there exists a sufficiently small positive constant τ such that

$$1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\pi}}\right)^{m_0} \;>\; p_0.$$

Let

$$p \;=\; 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\pi}}\right)^{m_0}.$$

Then

$$p \;\leq\; 1 - \left(\frac{1}{2} + \frac{\tau}{\sqrt{2\pi}}\right)^{m},$$

and it is easy to check that both τ and p can be totally determined by γ and θ. The proof is concluded by applying Corollary 3.3. □

3.6 Numerical experiments

The global rate analysis of the previous section roots in favor of using less directions in practice than the number usually required in deterministic direct search. In this section, we make use of a lower bound on this number to design an efficient strategy, that is shown to outperform its deterministic counterparts.

3.6.1 Practical satisfaction of the probabilistic descent property

Corollary 3.4 sheds a new light on the numerical testing we conducted at the beginning of this chapter. Remember that we tested Algorithm 3.1 by choosing $\mathfrak{D}_k$ as m independent random vectors uniformly distributed on the unit sphere in $\mathbb{R}^n$. This is a simple distribution to compute as such directions can be obtained by normalizing independent random vectors from the standard normal distribution in $\mathbb{R}^n$ [97]. In addition, the almost-sure global convergence of Algorithm 3.1 is guaranteed as long as $m > \log_2[1 - (\ln\theta)/(\ln\gamma)]$, as it can be concluded from Theorem 3.1. For example, when $\gamma = 2$ and $\theta = 0.5$, the algorithm converges with probability 1 when $m \geq 2$, even if such values of m are much smaller than the number of elements of the positive spanning sets with smallest cardinality in $\mathbb{R}^n$, which is $n + 1$ (41 in the cases tested in Section 3.2.2).

Towards optimality of the number of directions Given a unit vector $v \in \mathbb{R}^n$ and a number $\kappa \in [0, 1]$, it is easy to see that the event $\{\mathrm{cm}(\mathfrak{D}_k, v) \geq \kappa\}$ is the union of $\{\mathfrak{d}^\top v \geq \kappa\}$ and $\{\text{-}\mathfrak{d}^\top v \geq \kappa\}$, whose intersection has probability zero, and therefore

$$\mathbb{P}\big(\,\mathrm{cm}(\mathfrak{D}_k, v) \geq \kappa\big) \;=\; \mathbb{P}\big(\mathfrak{d}^\top v \geq \kappa\big) + \mathbb{P}\big(\text{-}\mathfrak{d}^\top v \geq \kappa\big) \;=\; 2\varrho,$$

ϱ being the probability of $\{\mathfrak{d}^\top v \geq \kappa\}$. One then sees that $\{\mathfrak{D}_k\}$ defined in this way is $(2\varrho, \kappa)$-descent. Given any constants γ and θ satisfying $0 < \theta < 1 < \gamma$, we can pick $\kappa > 0$ sufficiently small so that $2\varrho > (\ln\theta)/[\ln(\gamma^{-1}\theta)]$, similarly to what is stated in Corollary 3.4. Algorithm 3.1 then conforms to the theory presented in Sections 3.3 and 3.4. Moreover, the set $\{\mathfrak{d}, \text{-}\mathfrak{d}\}$ turns out to be *optimal* among all the sets $\mathfrak{D}$ consisting of 2 random vectors uniformly distributed on the unit sphere, in the sense that it maximizes the probability $\mathbb{P}\big(\,\mathrm{cm}(\mathfrak{D}, v) \geq \kappa\big)$ for each $\kappa \in [0, 1]$[1]. In fact, if $\mathfrak{D} = \{\mathfrak{d}_1, \mathfrak{d}_2\}$ with $\mathfrak{d}_1$ and $\mathfrak{d}_2$ uniformly distributed on the unit sphere, then

$$\mathbb{P}\big(\,\mathrm{cm}(\mathfrak{D}, v) \geq \kappa\big) \;=\; 2\varrho - \mathbb{P}\big(\{\mathfrak{d}_1^\top v \geq \kappa\} \cap \{\mathfrak{d}_2^\top v \geq \kappa\}\big) \;\leq\; 2\varrho,$$

and the maximal value 2ϱ is attained when $\mathfrak{D} = \{\mathfrak{d}, \text{-}\mathfrak{d}\}$ as already discussed. We tested the set $\{\mathfrak{d}, \text{-}\mathfrak{d}\}$ numerically (with $\gamma = 2$ and $\theta = 1/2$), and it performed even better than the set of 2 independent vectors uniformly distributed on the unit sphere (yet the difference was not substantial), which illustrates again our theory of direct search based on probabilistic descent.

Note that extensions of this reasoning to a number of polling directions greater than 2 constitutes a non-trivial geometrical problem (see [48] for a discussion on this topic, in a deterministic yet relevant setting).

3.6.2 A meaningful comparison between deterministic and randomized polling

The techniques described in the previous section, as well as the complexity theory we developed in Section 3.4, only apply when $\gamma > 1$. On the other hand, for deterministic

[1] As mentioned in [67], thanks are due to Professor Nick Trefethen for pointing out this property.

direct search based on positive spanning sets (PSSs), setting $\gamma = 1$ tends to lead to better numerical performance (see, for instance, [115]). In this sense, our introductory experiment of Section 3.2.2 appears biased in favor of probabilistic descent. To be fairer, we designed a new experiment by keeping $\gamma = 1$ when the sets of polling directions are guaranteed PSSs (which is true for the versions corresponding to $D_\oplus$, $Q\,D_\oplus$, and $Q_k\,D_\oplus$), while setting $\gamma > 1$ for direct search based on probabilistic descent. All the other parameters were selected as in Subsection 3.2.2.

In the case of direct search based on probabilistic descent, we pick now $\gamma = 2$ and $\gamma = 1.1$ as illustrations, and as for the cardinality m of $\mathfrak{D}_k$ we simply take the smallest integers satisfying $m > \log_2[1 - (\ln\theta)/(\ln\gamma)]$, which are 2 and 4 respectively. Table 3.3 presents the results of the redesigned experiment with $n = 40$. Table 3.4 shows what happened for $n = 100$. The data is organized in the same way as in Table 3.2.

We can see from the tables that direct search based on probabilistic descent still outperforms (for these problems) the direct-search versions using PSSs, even though the difference is not so considerable as in Table 3.2. We note that such an effect is even more visible when the dimension is higher ($n = 100$), which is somehow in agreement with the fact that (3.48) reveals a worst-case complexity in function evaluations of $\mathcal{O}(mn\epsilon^{-2})$, as this bound is more favorable than $\mathcal{O}(n^2\epsilon^{-2})$ for deterministic direct search based on PSSs when m is significantly smaller than n.

Table 3.3: Relative performance for different sets of polling directions ($n = 40$).

	$D_\oplus$	$Q\,D_\oplus$	$Q_k\,D_\oplus$	2 ($\gamma = 2$)	4 ($\gamma = 1.1$)
arglina	1.00	3.17	37.19	5.86	6.73
arglinb	34.12	5.34	32.56	1.00	2.02
broydn3d	1.00	1.91	5.96	2.04	3.47
dqrtic	1.18	1.36	28.32	1.00	1.48
engval1	1.05	1.00	16.44	2.29	2.89
freuroth	17.74	7.39	7.48	1.35	1.00
integreq	1.54	1.49	5.36	1.00	1.34
nondquar	1.00	2.82	8.02	1.37	1.73
sinquad	–	1.26	–	1.00	–
vardim	20.31	11.02	2.97	1.00	1.84

Table 3.4: Relative performance for different sets of polling directions ($n = 100$).

	$D_\oplus$	$Q\,D_\oplus$	$Q_k\,D_\oplus$	2 ($\gamma = 2$)	4 ($\gamma = 1.1$)
arglina	1.00	3.86	105.50	5.86	7.58
arglinb	138.28	107.32	106.23	1.00	1.99
broydn3d	1.00	2.57	12.07	1.92	3.21
dqrtic	3.01	3.25	–	1.00	1.46
engval1	1.04	1.00	43.00	2.06	2.84
freuroth	31.94	17.72	12.42	1.36	1.00
integreq	1.83	1.66	13.46	1.00	1.22
nondquar	1.18	2.83	23.15	1.00	1.17
sinquad	–	–	–	–	–
vardim	112.22	19.72	8.04	1.00	2.36

3.7 Conclusion and references for Chapter 3

By using randomly independently generated polling sets within a direct-search framework, one relaxes the assumptions on the polling directions, while allowing the use of smaller sets than in the classical, deterministic setting. Still, provided those sets have a minimum probability of containing one descent direction, we proved that the convergence properties of the deterministic case hold with probability one. The worst-case performance of the algorithm was also investigated: the associated results hold with overwhelming probability, and enlighten a potential gain in the worst-case number of function evaluations when using fewer directions than necessary for satisfying the deterministic assumptions.

An implementation based on uniformly distributed directions was proposed, and was found to outperform the deterministic variants based on a finite or infinite number of Positive Spanning Sets. This performance can be explained by the economy realized on the number of directions used per iteration. Indeed, the minimum number of vectors to use for guaranteeing (almost-sure) global convergence can be chosen independently of the problem dimension, contrary to the deterministic case. As a result, iteration cost in terms of calls to the objective is much lower, thereby compensating the possibly poor quality of the directions. In particular, the choice of two directions uniformly distributed in the unit sphere was investigated and shown to be the optimal strategy for common values of algorithmic parameters.

The material in this chapter was previously published in *SIAM Journal on Optimization* [67], with the exception of the lim result of Theorem 3.2 and the bound on the expected number of evaluations of Proposition 3.5 which we have introduced for the first time here in the thesis. We also mention that an implementation of Algorithm 3.1 based on the uniform generation technique was recently tested on an engineering problem [21].

Chapter 4

Trust-region methods based on probabilistic models for derivative-free optimization

The previous chapter provides both a convergence and a complexity analysis of a direct-search algorithm. The principles that are used for the probabilistic proofs can actually be applied to a wider range of methods, as long as appropriate elements of randomization have been identified. In particular, model-based algorithms can be studied in such a perspective, even though they significantly differ in nature from the class of direct-search schemes.

In this chapter, we develop a short study of derivative-free trust-region algorithms centered on the probabilistic case. After describing the deterministic features of such algorithms, tailored to model building via sampled values, in Section 4.1, we provide a convergence study of a derivative-free trust-region method relying on probabilistic models in Section 4.2. Such results are completed by novel probabilistic complexity results, that are established in Section 4.3 following the same process as in the direct-search case. We end this chapter by discussing the practicality of the considered properties in Section 4.4.

4.1 Deterministic derivative-free trust-region algorithms

Along with the direct-search methodology, model building is arguably one of the most famous techniques to address derivative-free optimization problems. The associated *model-based class* of algorithms relies on using sampled points to build a model of the objective function around the current iterate, typically by quadratic interpolation. Provided the resulting model is a sufficiently good approximation of the objective, it will resemble this function on a small neighborhood. The *trust-region* method is then particularly relevant to such a setting. It iteratively attempts to minimize a model constructed with previously computed function values within a trust-region, generally defined as a Euclidean ball of the type

$$B(x_k; \delta_k) = \{\, x_k + s \in \mathbb{R}^n : \|s\| \leq \delta_k \,\},$$

where x_k denotes the current iterate and δ_k the current trust-region radius. Trust-region algorithms have been widely studied in the derivative-based case [33], where models originated from Taylor expansions are common choices.

In DFO, the models are typically chosen as polynomials, that are built by fitting a sample set using interpolation or regression. Their quality is measured by the accuracy they provide relatively to a Taylor expansion. Models that are as accurate as first-order (resp. second-order) Taylor ones are then called fully linear (resp. fully quadratic) [37]. We precise these notions in the two following definitions.

Definition 4.1 *Given a function f continuously differentiable, $x \in \mathbb{R}^n$ and $\delta > 0$, a function $m : \mathbb{R}^n \to \mathbb{R}$ is called a* (κ_g, κ_f)-fully linear model *of f on $B(x, \delta)$, the ball of center x and radius δ, if it exists two positive constants κ_f and κ_g such that for all $x \in B(0, \delta)$,*

$$\begin{aligned} \|m(s) - f(x+s)\| &\leq \kappa_f\, \delta^2, \\ \|\nabla m(s) - \nabla f(x+s)\| &\leq \kappa_g\, \delta. \end{aligned}$$

Note that although linear polynomials can form fully linear models, quadratic functions are often preferred.

Definition 4.2 *Given a function f twice continuously differentiable, $x \in \mathbb{R}^n$ and $\delta > 0$, a function $m : \mathbb{R}^n \to \mathbb{R}$ is called a* $(\kappa_h, \kappa_g, \kappa_f)$-fully quadratic model *of f on $B(x, \delta)$, if it exists three positive constants κ_f, κ_g and κ_h such that for every $s \in B(0, \delta)$,*

$$\begin{aligned} \|m(s) - f(x+s)\| &\leq \kappa_f\, \delta^3, \\ \|\nabla m(s) - \nabla f(x+s)\| &\leq \kappa_g\, \delta^2, \\ \|\nabla^2 m(s) - \nabla^2 f(x+s)\| &\leq \kappa_h\, \delta. \end{aligned}$$

The construction of fully linear/quadratic models based on sampled sets raises a number of geometrical questions. Conn, Scheinberg and Vicente provided the first systematic approach to the subject of deterministic sampling geometry in [35, 36, 38], where

they specifically derived error bounds on polynomial interpolation and regression models, provided a certain quality of the sample sets (called *poisedness*) can be assessed. Later studies have then established that although the need of controlling geometry can be questionable for general purposes [55], model improvement (typically through generation of a better shaped sample set) is at least required whenever the size of the model gradient becomes small (a procedure known as the "criticality step", which ensures that the trust-region radius converges to zero) [105]. In this chapter, we will focus on a general, elementary trust-region method to serve as reference for our study. It will not involve the criticality step mentioned above, however the results can be extended to consider such a process. To lighten the notations and the upcoming proofs, we rather adopt this simplified setting, which is better suited for our purpose.

A typical derivative-free trust-region method can be shown to converge to a first-order stationary point, provided the models are fully linear and a *fraction of Cauchy decrease* (see Section 4.2) is satisfied by the trust-region step. In terms of complexity results, it can be established that such a derivative-free trust-region method will require $\mathcal{O}(\kappa_g^{-2}\,\epsilon^{-2})$ iterations and $\mathcal{O}(r\,\kappa_g^{-2}\,\epsilon^{-2})$ function evaluations to drive the minimum gradient norm under some threshold ϵ [60], with r being the number of points used to compute a model at an iteration. Since typical interpolation techniques ensure that $\kappa_g = \mathcal{O}(\sqrt{n})$ with $r = n + 1$, we recover a bound in $\mathcal{O}(n^2\,\epsilon^{-2})$, as for classical deterministic direct search. One can also design second-order globally convergent trust-region algorithms for DFO. To do so, one must rely on fully quadratic models at every iteration, while computing a step that satisfies both a fraction of Cauchy decrease and a *fraction of eigendecrease* (related to the potential negative curvature of the model). The criticality measure in the update formulas also has to be changed, from the norm of the model gradient to the maximum of the model gradient norm and the opposite of the minimum eigenvalue of the model Hessian. In that case, one can obtain a complexity bound in $\mathcal{O}\left(n^5\,\epsilon^{-3}\right)$ on the number of function evaluations needed to decrease this mixed criterion under the threshold ϵ. Such a result can be found in the thesis [73]. Note that contrary to the first-order one, it does not always match the derivative-based case [25].

The context of expensive function evaluations, endemic in DFO, often makes it unaffordable to construct a deterministic model that is guaranteed to be fully quadratic, as such a process requires $\frac{(n+1)(n+2)}{2}$ function evaluations. Practical approaches rely on considerably less points (but at least $n + 1$ to preserve fully linearity), and use the remaining degrees of freedom to minimize the norm of the model Hessian or its distance with the previous one. The most studied example is the minimum Frobenius norm update [34, 103], yet recent advances have proposed to apply the theory of sparse ℓ_1-recovery to build quadratic models based on random sampling [13]. Such models were proved to be fully quadratic even when considerably less points than $\frac{(n+1)(n+2)}{2}$ were used, depending on the sparsity pattern of the Hessian of the objective. Such findings called for a probabilistic analysis of derivative-free trust-region algorithms, which then led to the consideration of trust-region methods where the accuracy of the models is given with some positive probability. A study of this kind was performed in [14], and is described in the next section for first-order globally convergent trust-region methods.

4.2 Convergence of trust-region methods based on probabilistic models

Algorithm 4.1 describes the deterministic trust-region framework we will use throughout the chapter. Note that the trust-region radius plays a role similar to that of the step size parameter in Algorithm 2.1, while the quality of the models has the same importance than the quality of the polling sets.

Algorithm 4.1: Basic Derivative-Free Trust-Region framework

Define positive parameters η_1, η_2, $0 < \gamma_1 < 1 < \gamma_2$ and $0 < \delta_0 < \delta_{\max}$.
for $k = 0, 1, 2, \ldots$ **do**
  Approximate the function f in $B(x_k, \delta_k)$ by
  $m_k(x_k + s) = f(x_k) + g_k^\top s + \frac{1}{2} s^\top H_k s$.
  Compute s_k by approximately minimizing m_k in $B(x_k, \delta_k)$, and let

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

  If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$, and

$$\delta_{k+1} = \begin{cases} \min\{\gamma_2 \delta_k, \delta_{\max}\} & \text{if } \|g_k\| \geq \eta_2 \delta_k, \\ \gamma_1 \delta_k & \text{otherwise.} \end{cases}$$

  Otherwise, set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma_1 \delta_k$.
end

As mentioned in the previous section, if we suppose that the models m_k are fully linear at every iteration, it is possible to derive a (first-order) convergence analysis of Algorithm 4.1. In this section, we will present the results established by Bandeira, Scheinberg and Vicente [14] in a setting where the quality of the models is only required to be favorable with a given probability. Note that we slightly extend their framework by using two different parameters (namely γ_1 and γ_2) to update the trust-region radius, instead of using one and its inverse. As we will see, these parameters have a significant impact on the minimum probability with which our probabilistic properties need to be satisfied.

Algorithm 4.2 restates the method while conveniently introducing the appropriate random notations. Note that additional randomness is introduced at every iteration through the generation of the random model M_k (or, equivalently, the vector G_k and the symmetric matrix $\mathcal{H}_k$). This implies that the current point x_k, the trust-region step s_k and the trust-region radius δ_k also are of random nature. Thus, in the rest of this chapter, these random quantities will be denoted by $M_k, G_k, \mathcal{H}_k, X_k, S_k, \Delta_k$, while m_k, g_k, H_k, x_k, s_k and δ_k will indicate their respective realizations.

Algorithm 4.2: Derivative-Free Trust-Region based on Probabilistic Models

Define positive parameters η_1, η_2, $0 < \gamma_1 < 1 < \gamma_2$ and $0 < \delta_0 < \delta_{\max}$.
for $k = 0, 1, 2, \ldots$ **do**
Approximate the function f in $B(X_k, \Delta_k)$ by a randomly generated model $M_k(X_k + s) = f(X_k) + G_k^\top s + \frac{1}{2} s^\top \mathcal{H}_k s$.
Compute S_k by approximately minimizing M_k in $B(X_k, \Delta_k)$, and let

$$\rho_k \;=\; \frac{f(X_k) - f(X_k + S_k)}{M_k(X_k) - M_k(X_k + S_k)}.$$

If $\rho_k \geq \eta_1$, set $X_{k+1} = X_k + S_k$, and

$$\Delta_{k+1} \;=\; \begin{cases} \min\{\gamma_2 \Delta_k, \delta_{\max}\} & \text{if } \|G_k\| \geq \eta_2 \Delta_k, \\ \gamma_1 \Delta_k & \text{otherwise.} \end{cases}$$

Otherwise, set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma_1 \delta_k$.
end

4.2.1 Preliminary deterministic results

We place ourselves in the smooth context of Assumption 2.1 (i.e., f is continuously differentiable with Lipschitz continuous gradient) and assume that the models satisfy the following properties.

Assumption 4.1 *It exists $B_H > 0$ such that for every realization of Algorithm 4.2, the sequence of model Hessians satisfies*

$$\forall\, k, \quad \|H_k\| \;\leq\; B_H.$$

Assumption 4.2 *For all realizations of Algorithm 4.2, the step s_k of the k-th iteration is computed so that it satisfies a* fraction of Cauchy decrease, *i.e.*

$$m(x_k) - m(x_k + s_k) \;\geq\; \frac{\tau}{2} \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, \delta_k \right\}, \tag{4.1}$$

for some $\tau \in (0, 1)$, and with the convention that $\frac{\|g_k\|}{\|H_k\|} = \infty$ if $\|H_k\| = 0$.

Lemma 4.1 *For any realization of Algorithm 4.2, the series of trust-region radii is bounded as follows:*

$$\sum_{k=0}^{\infty} \delta_k^2 \leq \beta := \frac{\gamma_2^2}{1 - \gamma_1^2} \left[\frac{\delta_0^2}{\gamma_2^2} + \frac{f_0 - f_{\text{low}}}{\xi} \right], \tag{4.2}$$

with $f_0 = f(x_0)$ and

$$\xi = \eta_1 \eta_2 \frac{\tau}{2} \min\left\{ \frac{\eta_2}{B_H}, 1 \right\}.$$

As a result, $\lim_{k \to \infty} \delta_k = 0$.

Proof. We focus on the case where there are infinitely many iterations at which the trust-region radius is possibly increased. Let $\mathcal{K} \subset \mathbb{N}$ denote the set of such indexes.

For any $k \in \mathcal{K}$, we have by definition $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2 \, \delta_k$. Thus,

$$\begin{aligned} f(x_k) - f(x_k + s_k) &\geq \eta_1 \, (m_k(x_k) - m_k(x_k + s_k)) \\ &\geq \eta_1 \, \frac{\tau}{2} \min\left\{ \frac{\eta_2}{B_H}, 1 \right\} \eta_2 \, \delta_k^2 = \xi \, \delta_k^2 \end{aligned}$$

thanks to Assumption 4.2.

Consequently, if we sum on a finite set of consecutive very successful iterations, we obtain:

$$\xi \sum_{\substack{j \in \mathcal{K} \\ j \leq k}} \delta_j^2 \leq \sum_{\substack{j \in \mathcal{K} \\ j \leq k}} f(x_j) - f(x_{j+1}) \leq \sum_{j \leq k} f(x_j) - f(x_{j+1}) \leq f_0 - f(x_{k+1}) \leq f_0 - f_{\text{low}},$$

which leads to

$$\sum_{k \in \mathcal{K}} \delta_k^2 \leq \frac{f_0 - f_{\text{low}}}{\xi}.$$

We now denote by $\{k_i\}_{i \in \mathbb{N}}$ the very successful iterations, defining $k_0 = -1$ and $\delta_{-1} = \delta_0 / \gamma_2$. The sum $\sum_{k=0}^{\infty} \delta_k^2$ can thus be rewritten as follows:

$$\sum_{k=0}^{\infty} \delta_k^2 \; = \; \sum_{i=0}^{\infty} \sum_{k=k_i+1}^{k_{i+1}} \delta_k.$$

Besides, one has for each index i:

$$\delta_k \leq \gamma_2 \, (\gamma_1)^{k-k_i-1} \, \delta_{k_i}$$

for $k = k_i + 1, \ldots, k_{i+1}$. Thus,

$$\sum_{k=k_i+1}^{k_{i+1}} \delta_k^2 \; \leq \; \frac{\gamma_2^2}{1 - \gamma_1^2} \delta_{k_i}^2.$$

We finally obtain:

$$\sum_{k=0}^{\infty} \delta_k^2 \; \leq \; \frac{\gamma_2^2}{1 - \gamma_1^2} \sum_{i=0}^{\infty} \delta_{k_i}^2 \; \leq \; \frac{\gamma_2^2}{1 - \gamma_1^2} \left[\frac{\delta_0^2}{\gamma_2^2} + \frac{f_0 - f_{\text{low}}}{\xi} \right],$$

hence the result. □

We now provide an equivalent of Lemma 3.3 tailored to the trust-region case.

Lemma 4.2 *Consider a realization of Algorithm 4.2 and an index k for which the model m_k is (κ_g, κ_f)-fully linear on $B(x_k, \delta_k)$. Provided*

$$\delta_k \; < \; \min\left\{ \frac{\|g_k\|}{B_H}, \frac{\tau(1 - \eta_1)\|g_k\|}{4 \, \kappa_f}, \frac{\|g_k\|}{\eta_2} \right\} \tag{4.3}$$

holds at the k-th iteration, we have $x_{k+1} = x_k + s_k$ and $\delta_{k+1} = \min\{\gamma_2 \, \delta_k, \delta_{\max}\}$.

Proof. See [38, Proof of Lemma 10.6] for a proof that (4.3) implies that $\rho_k \geq \eta_1$ and thus that the iterate is updated. Since we also have $\|g_k\| \geq \eta_2 \, \delta_k$, the trust-region radius is increased. □

Whenever the models all satisfy the same fully linear property, the above lemmas are sufficient to derive a convergence analysis [37]. In the next section, we derive its probabilistic counterpart.

4.2.2 Probabilistically fully linear models and first-order properties

The key assumption to perform a convergence analysis of Algorithm 4.2 is that the random models exhibit fully linear properties with sufficiently high probability, as described below.

Definition 4.3 *We say that a sequence of random models $\{M_k\}$ is (p, κ_g, κ_f)-fully linear for a corresponding sequence $\{B(X_k, \Delta_k)\}$ if the events*

$$S_k \; = \; \{M_k \text{ is a } (\kappa_g, \kappa_f)\text{-fully linear model of } f \text{ on } B(X_k, \Delta_k)\}$$

satisfy the following submartingale-like condition

$$\mathbb{P}(S_0) \; \geq \; p, \quad \text{and} \quad \forall \, k \geq 1, \; \mathbb{P}\left(S_k | \sigma\left(M_0, \ldots, M_{k-1}\right)\right) \; \geq \; p.$$

Assumption 4.3 *The model sequence is a (p, κ_g, κ_f)-fully linear sequence, with $p \geq p_0$ and*

$$p_0 \; = \; \frac{\ln(\gamma_1)}{\ln\left(\gamma_1/\gamma_2\right)}.$$

As for Assumption 3.2 in the direct-search case, Assumption 4.3 is the key in establishing a suitable submartingale (with bounded increments) ensuring that the minimum gradient norm cannot stay bounded away from zero. Depending on the values of γ_1 and γ_2, p_0 need not necessarily be bigger than 0.5. This is an important point that was not considered in [14], although the results for $\gamma_1 = \gamma_2^{-1}$ easily extend to the general setting. This leads to the following convergence result.

Theorem 4.1 (see [14, Theorem 4.2]) *Consider Algorithm 4.1 applied to an objective function satisfying Assumptions 2.1 and 2.2. Suppose further that a random sequence of models is used throughout the method, and that said sequence satisfies Assumptions 4.1 to 4.3. Then, the sequence of random iterates produced by Algorithm 4.1 satisfies*

$$\mathbb{P}\left(\liminf_{k \to \infty} \|\nabla f(X_k)\| \; = \; 0 \right) \; = \; 1.$$

It is also possible to establish a lim-type result. Unlike in the direct-search case, no additional assumptions are required. The proof follows the lines of Section 3.3.4.

Theorem 4.2 ([14, Theorem 4.3]) *Under the assumptions of Theorem 4.1, the sequence of iterates also satisfies*

$$\mathbb{P}\left(\lim_{k \to \infty} \|\nabla f(X_k)\| \; = \; 0 \right) \; = \; 1.$$

4.3 Complexity study of trust-region methods based on probabilistic models

Following Section 3.4, it is possible to carry out a complexity analysis of a randomized trust-region framework based on probabilistically fully linear models. In the direct-search setting, we saw that Lemma 3.8 was the main argument leading to the global rate results of Section 3.4. It was based only on the two following elements of a realization of Algorithm 3.1:

1. if the k-th iteration is successful, then $f(x_k) - f(x_{k+1}) \geq \rho(\alpha_k)$ (in which case α_k is increased), and

2. if $\mathrm{cm}(D_k, -g_k) \geq \kappa$ and $\alpha_k < \varphi(\kappa \|g_k\|)$, then the k-th iteration is successful.

One can easily identify similar elements in a realization of the trust-region framework of Algorithm 4.2. Letting

$$\mathcal{K} \;=\; \{k \in \mathbb{N} : \rho_k \geq \eta_1 \text{ and } \|g_k\| \geq \eta_2 \, \delta_k\} \,,$$

one can find positive constants μ_1 and μ_2 such that:

1. if $k \in \mathcal{K}$, then $f(x_k) - f(x_{k+1}) \geq \mu_1 \, \delta_k^2$ and δ_k is increased (by definition of the method and Assumption 4.2 on the model decrease), and

2. if m_k is (κ_g, κ_f)-fully linear and $\delta_k < \mu_2 \|g_k\|$, then $k \in \mathcal{K}$ (see Lemma 4.2).

One sees that δ_k and $\mathcal{K}$ play the same roles as α_k and the set of successful iterations for the analysis of Section 3.4.

It is then straightforward to mimic the analysis of Section 3.4, by first bounding the number of iterations at which (κ_g, κ_f)-fully linear models appear.

Lemma 4.3 *Let Z_l be the indicator function of the random event*

$$\{ \; M_l \text{ is } (\kappa_g, \kappa_f) - \text{fully linear on } B(X_l, \Delta_l) \; \} \,,$$

and denote its realization by z_l. Consider a realization of Algorithm 4.2 and an index $k \geq 1$; one has

$$\sum_{l=0}^{k-1} z_l \leq \frac{\beta}{\mu^2} \max \left\{ \frac{\gamma_2^2 \mu^2}{\delta_0^2}, \frac{1}{\|\nabla f(\tilde{x}_k)\|^2} \right\} + p_0 \, k, \tag{4.4}$$

where

$$\mu = \frac{\zeta}{2(1 + \kappa_g \, \zeta)}, \;\; \zeta = \min \left\{ \frac{1}{\eta_2}, \frac{1}{B_H}, \frac{\tau(1 - \eta_1)}{4 \, \kappa_f} \right\};$$

and

$$\|\nabla f(\tilde{x}_k)\| = \inf_{0 \leq l \leq k} \|\nabla f(x_l)\|.$$

We can then derive the following complexity bounds, that can be viewed as equivalents of Corollaries 3.1 and 3.2 for trust-region methods.

Theorem 4.3 *Under Assumptions of Theorem 4.1, suppose that the model sequence is (p, κ_f, κ_g)-descent with $p > p_0$, with p_0 defined as in Assumption 4.3. Then, if one has*

$$k \;\geq\; \frac{2\,\beta\,\gamma_2^2}{(p-p_0)\,\delta_0^2}, \tag{4.5}$$

the minimum gradient norm $\|\nabla f(\tilde{X}_k)\|$ satisfies

$$\mathbb{P}\left(\|\nabla f(\tilde{X}_k)\| \;\leq\; \frac{\sqrt{2}\,\beta^{\frac{1}{2}}\,(p-p_0)^{\frac{1}{2}}}{\mu}\,\frac{1}{\sqrt{k}}\right) \;\geq\; 1 - \exp\left[-\frac{(p-p_0)^2}{8\,p}k\right]. \tag{4.6}$$

Theorem 4.4 *Under Assumptions of Theorem 4.1, suppose that the model sequence is (p, κ_f, κ_g)-descent with $p > p_0$, with p_0 defined as in Assumption 4.3. Then, provided*

$$\epsilon \;<\; \frac{\delta_0}{\mu\,\gamma_2}, \tag{4.7}$$

the (random) first iteration index K_ϵ for which $\|\nabla f(X_{K_\epsilon+1})\| < \epsilon$ satisfies

$$\mathbb{P}\left(K_\epsilon \;\leq\; \left\lceil \frac{2\,\beta}{(p-p_0)\,\mu^2\,\epsilon^2} \right\rceil\right) \;\geq\; 1 - \exp\left[-\frac{\beta\,(p-p_0)\,\delta^2}{4\,p\,\mu^2\,\epsilon^2}\right]. \tag{4.8}$$

Note that by the definition of μ, one obtains a bound on the number of iterations is of order of $\mathcal{O}\left(\kappa_g^{-2}\,\epsilon^{-2}\right)$ with overwhelming probability.

4.4 Practical insights on probabilistic models

Generating fully linear models in a deterministic fashion essentially requires $\mathcal{O}(n)$ function evaluations corresponding to a poised sample set. In a randomized setting, as noted in [14], it is possible to use random sample sets, e.g. following a standard Gaussian distribution. By generating n vectors of this type (plus an additional zero vector to use the value of the objective at the current point), one can bound the probability of the resulting sample set to be Λ-*poised* [38]. In that setting, there exist positive constants with respect to which the sequence of models built from the sample sets is probabilistically fully linear.

Although this result roots in favor of random models, their use has been surprisingly limited. In practice, most methods based on random models aim to address stochastic or noisy optimization problems, and it is generally assumed that the lack of quality of the models come from the inaccuracy of the function values rather than the sampling strategy. As a result, some instances of probabilistically fully-linear models can be obtained for specific cases of noisy functions [79] or evaluations subject to computational failures [30], but the size of the sample set to be used is never taken to be less than the

minimal number of points needed to build a (deterministic) fully linear model. To the best of our knowledge, only the presence of sparsity in the Hessian matrix has been proved to have an impact on the sample size. Indeed, it has been shown that random sample sets of $\mathcal{O}\left(n\left(\log n\right)^4\right)$ vectors could yield fully quadratic interpolations models. We refer to [13] and [14, Section 6.2] for further discussion on these sparsity aspects.

Nevertheless, it should be emphasized that the convergence theory derived in Section 4.2 sheds a new light on the issue of bad quality models in derivative-free trust-region methods. Indeed, by considering that every model generated by the algorithm has a minimum probability of being fully linear, one can still obtain theoretical guarantees. Therefore, one could apply the previous analysis on an algorithm that does not systematically check the quality of the model and still be able to certify its convergence (almost surely).

4.5 Conclusion and references for Chapter 4

In derivative-free optimization, the applicability of trust-region algorithms is tailored to the quality of the models that are iteratively constructed, generally by means of a polynomial approximation. When such a Taylor-like accuracy is ensured with a sufficiently high probability, convergence and complexity analyses can still be derived, yielding theoretical results that compare to the deterministic case. The similarities between the corresponding analysis and the one presented in Chapter 3 are not incidental: they illustrate the strength of the probabilistic reasoning that is applied in both cases.

Unlike the direct-search case, no practical implementation has been proposed in the general case that would satisfy the desirable property at a certifiably lower cost. Still, techniques relying on the same order of expense than for the deterministic setting have been proposed. The corresponding implementations may then rely on a simplified algorithmic framework, which might ease their use by a non-specialist.

The convergence theory of this chapter was presented by Bandeira et al [14], however we point out that the complexity analysis was not derived until the work described in Chapter 3 was completed. In [67], it was indeed provided a sketch of the application of the probabilistic complexity proof technique to a wider class of derivative-free methods, which we further developed in this chapter.

Chapter 5

Probabilistic feasible descent techniques for bound-constrained and linearly-constrained optimization

Most optimization problems appear under a constrained form, in that the variables on which the optimization process is applied are subject to one or several constraints, typically due to physical or budget limitations. The resulting problems are likely to become harder to solve, as one must take feasibility (i.e., satisfaction of the constraints) into account, and consider a trade-off between feasibility and minimization of the objective. In the case of simple constraints such as positivity of the variables, it is common to require that only feasible points are considered for function evaluation. In that case, the possibilities for exploring the variable space can be significantly reduced.

In this chapter, we detail several probabilistic strategies to handle simple constraints on the variables, namely bounds on those variables and linear equalities, within a direct-search framework. We open the chapter by Section 5.1, that shortly introduces the treatment of such constrained problems in the deterministic direct-search literature. We then focus on the bound-constrained case in Section 5.2, where we present two strategies based upon random variants on the deterministic setting, and derive the associated probabilistic analysis. Section 5.3 is dedicated to linear equality constraints, and enlightens the similarities between such a context and the unconstrained one from a probabilistic point of view. An implementation of the resulting method is proposed in Section 5.4, where it is favorably compared to a built-in MATLAB function.

5.1 Handling bounds and linear constraints in direct-search methods

When no constraints are enforced on the variables, the polling directions used in a direct-search method must provide good approximations of all vectors in the space: this is guaranteed by assuming that the polling vectors span the space by nonnegative linear combinations, and it is a key argument to derive suitable global convergence properties of the methods. Under the presence of constraints, the properties of such directions are closely related to the shape of the feasible domain. As a result, the quality of the polling sets at a particular point is generally based on the nearby constraints. When the boundary of the feasible domain is sufficiently simple, it is possible to design polling strategies such that there always exists a direction along which decrease in the function value is possible while remaining in the feasible domain [75].

Bound constraints are a classical example of such a setting. Direct-search methods tailored to these constraints have been widely studied [58, 74, 81, 87], mostly basing themselves on moves along the coordinate axes. Interestingly, this still forms the state-of-the-art polling choice for solving bound-constrained problems. To the best of our knowledge, no deterministic direct-search method that intends to exploit the geometry of bound constraints has avoided considering this type of directions. As for general linear constraints, they are often dealt with through an active-set approach based on identifying nearby constraints, corresponding to a portion of the feasible domain boundary that is close to the current iterate. Although such techniques are sensitive to degeneracy [5] and might require the use of tools from computational geometry [80], they have been shown to perform quite well in practice, while being supported by a theoretical analysis extending the unconstrained case. Indeed, the active-set strategies iteratively define cones that correspond to sets of feasible directions. The generators of such cones thus positively span feasible directions, which makes them a suitable choice for polling [77, 80, 82].

Most of the above references consider problems with linear inequality constraints, and rely on the active-set identification properties of the direct-search optimization process to eventually identify the constraints that are active at a stationary point [83]. Still, specific treatment of linear equality constraints has also been proposed, one way being to reformulate the problem so that those constraints are removed and (possibly) fewer variables are actually involved in the minimization process [9, 51]. It can be argued that this reformulation has a negative practical impact, in that it can give rise to a less separable problem: this is especially true for algorithms that transform the equality constraints into two additional inequality constraints [77]. Another possibility is to design the algorithm to work in the null space of the linear equality constraints, which is also equivalent to solving the problem in a lower-dimensional subspace [85].

All the aforementioned algorithms involve deterministic generation of the polling sets, in that those only depend on the constraints currently classified as approximately active. Even when coupled with global exploration strategies [46, 112, 113], the polling process must ensure some form of covering of the feasible descent area, either through deterministic generation or by a density argument (which is also a way to handle arbitrary

constraints). This may represent a significant expense in terms of function evaluations, possibly even higher than using a positive spanning set at every iteration. Therefore, it is natural to ask whether probabilistic variants of direct-search methods can be constructed to tackle linearly-constrained problems, albeit through dedicated treatment of the different categories of such constraints.

Algorithm 5.1: Direct Search based on Probabilistic Feasible Descent (DSPFD)

Choose an initial point $x_0 \in \mathbb{R}^n$, as well as $0 < \theta < 1 \leq \gamma$, $0 < \alpha_0 < \alpha_{\max}$.
Define a *forcing function* $\rho : \mathbb{R}^+ \to \mathbb{R}^+$.
for $k = 0, 1, 2, \ldots$ **do**
  Poll Step
  Compute a set of random independent vectors $\mathfrak{D}_k$.
  If there exists $\mathfrak{d} \in \mathfrak{D}_k$ such that $X_k + \mathcal{A}_k \mathfrak{d}$ **is a feasible point** and

$$f(X_k + \mathcal{A}_k \mathfrak{d}) - f(X_k) \ < \ -\rho\left(\mathcal{A}_k \|\mathfrak{d}\|\right),$$

  declare the iteration *successful* and set $X_{k+1} = X_k + \mathcal{A}_k \mathfrak{d}$;
  Otherwise declare the iteration as *unsuccessful* and set $x_{k+1} = x_k$.
  Step Size Update
  If the iteration is successful, (possibly) increase the step size by setting $\mathcal{A}_{k+1} = \min\{\gamma \mathcal{A}_k, \alpha_{\max}\}$;
  Otherwise, decrease the step size by setting $\mathcal{A}_{k+1} = \theta \mathcal{A}_k$.
end

Algorithm 5.1 presents the basic method of our analysis. Note that its only difference with Algorithm 3.1 is that it enforces feasibility of all the iterates. One may notice that this feasibility requirement does not appear when the polling sets are computed. Even though the upcoming theoretical results will rely on directions that generate feasible trial points, the method may attempt to evaluate f at infeasible points (in which case infeasibility is detected prior to the function call and the evaluation is not considered).

5.2 Probabilistic polling strategies for bound-constrained problems

In this section, we consider that only bound constraints are present. The corresponding problem is

$$\begin{cases} \min & f(x) \\ s.t. & l \leq x \leq u. \end{cases} \tag{5.1}$$

Our objective is to apply a direct-search framework such as Algorithm 5.1 in order to generate a sequence of feasible iterates $\{x_k\}$ converging towards a first-order stationary point of problem (5.1). Since feasibility has to be maintained throughout the iterates,

we need to identify the possible bounds of concern at every iteration, i.e., those that may prevent a move in the corresponding directions. Once this has been done, it is possible to adapt the polling strategy (i.e., the polling set) to cope with these "active" constraints.

Such an approach follows what has been proposed in the direct-search literature for general linear inequality constraints. In the case of bound constraints, however, the situation is simplified as the directions of interest are the coordinate vectors and their negatives. This is a key feature in the deterministic case, that determines an *ad hoc* polling strategy. Our objective is to build on this technique to generate feasible descent directions in probability.

5.2.1 The coordinate set and its associated feasible descent properties

We begin our analysis of bound-constrained problems by looking in detail at the properties of the *coordinate set*, which we recall is given by

$$D_{\oplus} = \{e_1, \ldots, e_n, \text{-}e_1, \ldots, \text{-}e_n\}\,.$$

The directions in that set will be called *coordinate vectors* or opposite coordinate vectors. $D_{\oplus}$ is a natural choice for polling if one aims to conform the search directions to the geometry of bound constraints. Besides, the fact that $D_{\oplus}$ is a PSS makes it a descent set even when no bounds actually intervene at an iteration [75, 81].

In order to study the convergence properties of direct-search schemes relying on $D_{\oplus}$ for solving bound-constrained problems, the cones generated by coordinate vectors are of critical importance. Given any cone K generated by a subset of $D_{\oplus}$, the projection of $v \in \mathbb{R}^n$ onto K, denoted by v_K is given component-wise as follows:

$$\forall\, i \in \{1, \ldots, n\}\,, \quad [v_K]_i \;=\; \begin{cases} [v]_i & \text{if } [v]_i > 0 \text{ and } e_i \in K, \\ [v]_i & \text{if } [v]_i < 0 \text{ and } \text{-}e_i \in K, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

We state the result that is of most interest in Lemma 5.1, which is a mild generalization of [75, Proposition 8.1].

Lemma 5.1 *Consider a cone K generated by a subset of $D_{\oplus}$, and a vector v such that its projection v_K on K is not the zero vector. Then,*

$$\max_{d \in D_{\oplus}} \frac{d^\top v}{\|d\|\|v_K\|} \;\geq\; \frac{1}{\sqrt{n}}, \tag{5.3}$$

and this maximum is attained at a generator $u \in D_{\oplus}$ of K, for which we have

$$u^\top v \;=\; \|v_K\|_\infty \;=\; \max_{1 \leq i \leq n} |[v_K]_i|\,. \tag{5.4}$$

Lemma 5.1 implies that for any cone K being positively spanned by a subset of the coordinate vectors, those same vectors can provide information on the largest component of the vector v_K through their scalar product with the vector v. This largest component may even be the largest of the original vector v, which increases further the quality of this information. Such a remarkable feature can be explained by the fact that $D_\oplus$ can be defined (not in a unique way) as the union of two orthonormal, opposite linear bases. In order to use other types of polling sets, we would like to express the properties of $D_\oplus$ in a more general fashion.

Consider a point $l \leq x_k \leq u$, and a step size $\alpha_k > 0$, corresponding to the k-th iteration of a realization of Algorithm 5.1. We define the index sets of the *free bound constraints at* (x_k, α_k) by

$$I^+(x_k, \alpha_k) \;=\; \{i \,|\, [x_k]_i + \alpha_k \leq u_i\} \quad \text{and} \quad I^-(x_k, \alpha_k) \;=\; \{i \,|\, l_i \leq [x_k]_i - \alpha_k\}\,. \tag{5.5}$$

For any index of those sets, a displacement from x_k along the i-th coordinate direction (either in the direction of $e_i, -e_i$ or both) with a step size less than or equal to α_k yields a feasible point. This is of considerable interest given that Algorithm 5.1 only considers steps of length α_k. Note that the analysis can be extended by replacing α_k in (5.5) by any sequence $\{\varepsilon_k\}_k$ such that $\varepsilon_k = o(\alpha_k)$ whenever k goes to infinity. Our study follows the practical choice suggested in [80].

The *approximate tangent cone* at (x_k, α_k) is then given by

$$T_k \equiv T(x_k, \alpha_k) \;=\; \text{pspan}\Big(\{e_i\}_{i \in I^+(x_k,\alpha_k)} \,,\; \{\text{-}e_i\}_{i \in I^-(x_k,\alpha_k)} \Big)\,. \tag{5.6}$$

Such a cone is particularly useful, since we know that it contains feasible directions (at least its generators).

We also define the *approximate normal cone* at (x_k, α_k) as follows:

$$N_k \equiv N(x_k, \alpha_k) \;=\; \text{pspan}\Big(\{e_i\}_{i \notin I^+(x_k,\alpha_k)} \,\cup\, \{\text{-}e_i\}_{i \notin I^-(x_k,\alpha_k)} \Big)\,. \tag{5.7}$$

N_k is the polar cone to T_k, which means that for any vector $v \in \mathbb{R}^n$ there exists a decomposition $v = v_{T_k} + v_{N_k}$ with $v_{T_k} \in T_k$, $v_{N_k} \in N_k$ and $v_{T_k}^\top v_{N_k} = 0$ [96]. Such a decomposition has been widely used in direct-search schemes addressing linear or bound constraints [75, 81].

The property of interest for our analysis relates only to the approximate tangent cone, as stated in Definition 5.1.

Definition 5.1 *Let $x_k \in \mathbb{R}^n$ such that $l \leq x_k \leq u$ and $\alpha_k > 0$ such that $[-\nabla f(x_k)]_{T_k} \neq 0$ if $T_k \neq \emptyset$, where T_k is defined following* (5.6).
A set of vectors D is called a κ-feasible descent set *for the problem* (5.1) *at (x_k, α_k) if*

$$\max_{\substack{d \in D \\ l \leq x_k + \alpha_k d \leq u}} \frac{d^\top\, [-\nabla f(x_k)]}{\|d\| \left\| [-\nabla f(x_k)]_{T_k} \right\|} \;\geq\; \kappa, \tag{5.8}$$

where $\kappa \in (0, 1)$.

By convention, one can consider a set as κ-feasible descent whenever the cone T_k is empty.

Given the result of Lemma 5.1, an obvious choice for such a polling set consists in using the coordinate vectors appearing in (5.6) (note that the size of this set can theoretically vary between 0 and $2\,n$). By considering all vectors from $D_{\oplus}$, one thus ensures both feasibility and descent when the step size is sufficiently small, while simultaneously covering all possibilities in terms of approximate tangent cone [75].

Proposition 5.1 enlightens the strength of the coordinate set while dealing with bound-constrained problems, and motivates further its extensive use within the direct-search community.

Proposition 5.1 *Let x_k and α_k be defined as in Definition 5.1. Then, the set $D_{\oplus}$ is a $\frac{1}{\sqrt{n}}$-feasible descent set, independently of the value of T_k.*

The coordinate set thus provides strong guarantees in terms of descent; its main drawback lies in its size, as using $D_{\oplus}$ in a direct-search method may result in $2\,n$ evaluations per iteration. Since it has been shown than using (much) less than $2\,n$ directions can still ensure convergence in probability in the unconstrained case, a natural question to ask is whether similar properties can be derived for the bound-constrained setting. We provide two possible answers in the next sections, by introducing polling rules that exhibit a (probabilistic) behavior similar to that of $D_{\oplus}$.

5.2.2 A probabilistic technique based on the coordinate set

Let $\mathcal{T}_k = T(X_k, \mathcal{A}_k)$ and T_k denote the approximate tangent cones in Algorithm 5.1 and one of its realizations, respectively. One can make use of such random sets to propose a probabilistic equivalent of Definition 5.1, as described below.

Definition 5.2 *Under the same considerations as in Definition 5.1, let $\{\mathfrak{D}_k\}$ be a sequence of randomly, independently generated sets, and $(\kappa, p) \in (0,1)^2$. Let*

$$E_k \;=\; \{\; \mathfrak{D}_k \text{ is a } \kappa\text{−feasible descent set for (5.1) at } (x_k, \alpha_k) \;\}\,. \tag{5.9}$$

The sequence $\{\mathfrak{D}_k\}$ is called a (p,κ)-feasible descent sequence *for problem* (5.1) *if*

$$\mathbb{P}\,(E_0) \;\geq\; p \quad \textit{and} \quad \forall\, k \geq 1,\; \mathbb{P}\,(E_k \mid \sigma\,(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1})) \;\geq\; p. \tag{5.10}$$

A trivial way of satisfying such a property is the deterministic choice $\mathfrak{D}_k = D_{\oplus}$ (or equivalently, a sequence of polling sets randomly independently drawn within a singleton of possibilities, namely $\{D_{\oplus}\}$). Our objective is to find sets, possibly of smaller size, that satisfy a similar probabilistic relation.

Our first proposal is a simple randomization of the deterministic strategy, that consists in randomly selecting a sample of the coordinate directions to serve as polling directions. This sample need only be drawn in the set of generators of the approximate tangent cone in order to ensure a non-zero probability of selecting a direction of feasible descent.

Suppose that the approximate tangent cone $\mathcal{T}_k$ at iteration k is generated by b_k vectors from the coordinate set $D_\oplus$, with $0 < b_k \leq 2\,n$. Then, by randomly uniformly selecting $s_k < b_k$ of those directions as the polling set $\mathfrak{D}_k$, the results of Section 5.2.1 imply that the probability of having a $\frac{1}{\sqrt{n}}$-feasible descent set is at least of s_k/b_k. Recall that when $b_k = 0$, no move of length α_k is possible in any of the coordinate directions, and since it implies that $[-\nabla f(X_k)]_{\mathcal{T}_k} = 0$, we may ignore it for the analysis by considering that the feasible descent property is satisfied no matter the value of the polling set. In practice, however, it is often useful to try to compute a step along the directions within the approximate normal cone in that case, ensuring that the corresponding step size does not go below some minimum value.

For any $p_0 \in (0,1)$, it is then easy to determine the minimum number of directions required to satisfy

$$\frac{s_k}{b_k} > p_0 \quad \Leftrightarrow s_k \geq \left\lceil \frac{b_k \, \ln \theta}{\ln(\gamma^{-1}\,\theta)} \right\rceil .$$

Given that the tangent cone is, in the worst case, generated by $2\,n$ directions, randomly choosing $\lceil 2\,n\,p_0 \rceil$ coordinate directions at most (only when it is possible) would imply that the set sequence is a feasible descent sequence as defined below.

Proposition 5.2 *Let $p_0 \in (0,1)$ and $\mathfrak{D}_k \subset D_\oplus$ be chosen as a subset of $s_k > \lceil b_k\,p_0 \rceil$ columns within the generators of $\mathcal{T}_k$ whenever $\mathcal{T}_k \neq \emptyset$, independently drawn at every iteration. Then, $\{\mathfrak{D}_k\}$ is a $\left(p, \frac{1}{\sqrt{n}}\right)$-feasible descent set sequence for any $p > p_0$.*

The advantages and disadvantages of this approach are clear from by Proposition 5.2 and the above discussion. On the one hand, by generating fewer directions than those actually necessary to generate the entire approximate tangent cone, we might not consider the one satisfying (5.3). On the other hand, such a direction may not be the only feasible descent one. Moreover, by using a sample of the directions in $D_\oplus$, the cost of a full poll step is cheaper than in a standard coordinate search method.

5.2.3 Using subspaces to reduce the number of directions

Our second technique is designed to decrease further the size of the polling sets. The underlying idea is based on the results from Chapter 3, in which we have seen that direct search using randomly generated directions in $\mathbb{R}^n$ can rely on a number of directions that does not depend on the dimension of the space. Therefore, it seems possible to reduce the minimal number of polling directions whenever the approximate tangent cone contains a subspace.

Let $\mathcal{T}_k^s$ denote the *lineality space* of the approximate tangent cone $\mathcal{T}_k$, i.e. the subspace of highest dimension contained in this cone. We then have the following decomposition of $\mathcal{T}_k$ into a subspace and a cone that are orthogonal:

$$\mathcal{T}_k \;=\; \mathcal{T}_k^s + \mathcal{T}_k^c .$$

where $\mathcal{T}_k^s$ is linearly spanned by a subset $\{e_i\}_{i\in I_k^s}$ of the coordinate vectors, and $\mathcal{T}_k^c$ is a cone positively spanned by $\{e_i\}_{i\in I_k^{c+}} \cup \{\text{-}e_i\}_{i\in I_k^{c-}}$, with

$$I_k^{c+} = I_k^+ \setminus I_k^s, \quad I_k^{c-} = I_k^- \setminus I_k^s.$$

In what follows, T_k^s and T_k^c will represent realizations of $\mathcal{T}_k^s$ and $\mathcal{T}_k^c$.

The first part of our polling strategy will consist in using random directions in the lineality space $\mathcal{T}_k^s$, together with generators of the cone $\mathcal{T}_k^c$. For such vectors, one easily establishes the following property thanks to the definition of the projection over $\mathcal{T}_k$.

Lemma 5.2 *Under the definitions above, suppose that $\mathcal{T}_k \neq \emptyset$ and consider a direction d belonging to*

$$\mathcal{T}_k^s \bigcup \{e_i\}_{i\in I_k^{c+}} \bigcup \{\text{-}e_i\}_{i\in I_k^{c-}}. \tag{5.11}$$

Then, for any vector v,

$$d^\top v \;=\; d^\top [v]_{\mathcal{T}_k}.$$

Our aim is to reach a quality of the polling sets which is comparable with the one we obtain by using directions from $D_\oplus$. To do so, it suffices to guarantee that we can ensure descent in the lineality space whenever needed, as shown in the following lemma.

Lemma 5.3 *Consider a realization of Algorithm 5.1 and let D be a set of vectors in $\mathbb{R}^n$ such that $D \cap T_k^s$ is κ-descent in the subspace T_k^s, i.e., for any nonzero vector $w \in T_k^s$,*

$$\max_{d\in D\cap T_k^s} \frac{d^\top w}{\|d\|\|w\|} \;\geq\; \kappa.$$

Suppose further that D contains all generators of the cone T_k^c.

In that case, for any $v \in T_k$, $\|v\| \neq 0$, we have

$$\max_{d\in D} \frac{d^\top v}{\|d\|\|[v]_{T_k}\|} \;\geq\; \frac{\kappa}{\sqrt{n}}. \tag{5.12}$$

Proof. One must distinguish two cases, namely

$$\max_{d\in D_\oplus} \frac{d^\top v}{\|d\|\,\|\,[v]_{T_k}\,\|} \;=\; \max_{d\in D_\oplus\cap T_k^s} \frac{d^\top v}{\|d\|\|\,[v]_{T_k}\,\|} \tag{5.13}$$

and

$$\max_{d\in D_\oplus} \frac{d^\top v}{\|d\|\,\|\,[v]_{T_k}\,\|} \;=\; \max_{d\in D_\oplus\cap T_k^c} \frac{d^\top v}{\|d\|\|\,[v]_{T_k}\,\|}. \tag{5.14}$$

In the second case, since D contains the generators of T_k^c that belong to $D_\oplus$, it is clear that we have

$$\max_{d\in D} \frac{d^\top v}{\|d\|\,\|\,[v]_{T_k}\,\|} \;\geq\; \max_{d\in D_\oplus\cap T_k^c} \frac{d^\top v}{\|d\|\|\,[v]_{T_k}\,\|} \;\geq\; \frac{1}{\sqrt{n}}. \tag{5.15}$$

In the first case, since $D \cap T_k^s$ is κ-descent, we have

$$\max_{d \in D \cap T_k^s} \frac{d^\top [v]_{T_k^s}}{\|d\| \|[v]_{T_k^s}\|} \;\geq\; \kappa. \tag{5.16}$$

In addition, for any $d \in D \cap T_k^s$, we have $d^\top v = d^\top [v]_{T_k^s}$ and

$$\frac{\|[v]_{T_k^s}\|}{\|v_{T_k}\|} \;\geq\; \frac{1}{\sqrt{n}},$$

by definition of the cone T_k. This leads to

$$\max_{d \in D} \frac{d^\top v}{\|d\| \, \|[v]_{T_k}\|} \;\geq\; \max_{d \in D_\oplus \cap T_k^s} \frac{d^\top v}{\|d\| \, \|[v]_{T_k}\|} \;\geq\; \frac{1}{\sqrt{n}} \max_{d \in D_\oplus \cap T_k^s} \frac{d^\top v}{\|d\| \, \|[v]_{T_k^s}\|} \;\geq\; \frac{\kappa}{\sqrt{n}}. \tag{5.17}$$

Putting (5.15) and (5.17) yields the desired result. □

From Section 3.5, we know how to generate directions in the lineality space so that they have a probability p of being of κ-descent with $\kappa = \frac{\tau}{\sqrt{n}}$. In Section 5.2.2, we have seen a way to determine the size of the random sample of the remaining generators of T_k we have to draw, so as to guarantee the same probability of drawing any direction. As a result, we obtain the following property for our strategy.

Proposition 5.3 *Let $p_0 = \ln(\theta)/\ln(\gamma^{-1}\,\theta)$. Consider Algorithm 5.1 and let $\{\mathfrak{D}_k\}$ be its sequence of polling sets. Then, there exist s, that depends solely on θ and γ, and a sequence $\{s_k^c\}$, depending on p_0 and the number of generators of $\mathcal{T}_k^c$, such that if every $\mathfrak{D}_k$ is generated by taking s unit vectors uniformly in T_k^s and s_k^c generators of $\mathcal{T}_k^c$, the corresponding sequence is $(p, \frac{\tau}{n})$-feasible descent set for some $p \geq p_0$ and $\tau \in (0, \sqrt{n}]$ independent of the problem dimension.*

Proof. From Corollary 3.4, we know that a set formed of

$$s \;>\; \log_2\left(1 - \frac{\ln\theta}{\ln\gamma}\right)$$

generated in the intersection of the unit sphere with $\mathcal{T}_k^s$ is $\tau/\sqrt{n}$-descent with a probability $p \geq p_0$, where p and τ do not depend on n, provided the projection of the negative gradient onto this subspace is not zero.

Meanwhile, if we sample $s_k^c = \lceil p\, b_k^c \rceil$, where b_k^c is the number of generators of T_k^c, we know that these vectors form a $\frac{1}{\sqrt{n}}$-descent set with a probability $1/\sqrt{n}$.

As the two sets are independent, their union $\mathfrak{D}_k$ has a probability at least p of being in the assumptions of Lemma 5.3, with $\kappa = \frac{\tau}{\sqrt{n}}$. □

5.2.4 Theoretical properties of the probabilistic variants

This section presents a general convergence analysis of Algorithm 5.1 applied to bound-constrained problems. Our convergence result will involve the following criticality measure:

$$\chi(x) \;=\; \max_{\substack{l \leq x+w \leq u \\ \|w\| \leq 1}} w^\top \, [-\nabla f(x)] \, . \tag{5.18}$$

It is well-known that χ is a continuous, non-negative function of x, that equals 0 at first-order stationary points of the problem (5.1).

In addition to the assumptions of Chapter 3 regarding the boundedness and the smoothness of the objective function, we will also need the gradient to be bounded in norm over the feasible domain. This is the sense of the following assumption, which is standard in direct-search algorithms applied to linearly-constrained problems.

Assumption 5.1 *The gradient of the objective function is bounded in norm on the feasible set* $\{ \, l \leq x \leq u \, \}$*, i.e.,*

$$\exists \, B_g > 0, \; \forall x \in \{l \leq x \leq u\}, \quad \|\nabla f(x)\| \;\leq\; B_g.$$

Our analysis begins with the following key lemma on unsuccessful iterations.

Lemma 5.4 *Consider a realization of Algorithm 5.1 applied to problem* (5.1)*. Suppose that Assumptions 2.1 and 5.1 hold, and that* D_k *is a* κ*-feasible descent set at* (x_k, α_k) *with unit directions, with* k *being the index of an unsuccessful iteration. Then,*

$$\left\| [-\nabla f(x_k)]_{T_k} \right\| \;\leq\; \frac{1}{\kappa} \left(\frac{\nu_g}{2} \alpha_k + \frac{\rho(\alpha_k)}{\alpha_k} \right) \tag{5.19}$$

and

$$\chi(x_k) \;\leq\; \left[\frac{\nu_g}{2\,\kappa} + \sqrt{n}\, B_g \right] \alpha_k + \frac{\rho(\alpha_k)}{\kappa \, \alpha_k}. \tag{5.20}$$

Proof. Let d_k be a feasible direction in D_k satisfying

$$\frac{d_k^\top \, [-\nabla f(x_k)]}{\|d_k\| \, \| \, [-\nabla f(x_k)]_{T_k} \, \|} \;\geq\; \kappa.$$

By a first-order Taylor expansion of $f(x_k + \alpha_k \, d_k)$, one has

$$\begin{aligned} -\rho(\alpha_k) \;&\leq\; f(x_k + \alpha_k \, d_k) - f(x_k) \\ &\leq\; -\kappa \; \alpha_k \left\| [-\nabla f(x_k)]_{T_k} \right\| + \frac{\nu_g}{2} \alpha_k^2, \end{aligned}$$

thus (5.19) holds.

Besides, we have that

$$\begin{aligned}
\chi(x_k) &= \max_{\substack{l \leq x_k + w \leq u \\ \|w\| \leq 1}} w^\top \, [-\nabla f(x_k)] \\
&= \max_{\substack{l \leq x_k + w \leq u \\ \|w\| \leq 1}} \left(w^\top \, [-\nabla f(x_k)]_{T_k} + \left([w]_{T_k} + [w]_{N_k} \right)^\top \, [-\nabla f(x_k)]_{N_k} \right), \\
&\leq \max_{\substack{l \leq x_k + w \leq u \\ \|w\| \leq 1}} \left([w]^\top \, [-\nabla f(x_k)]_{T_k} + [w]_{N_k}^\top \, [-\nabla f(x_k)]_{N_k} \right),
\end{aligned}$$

by property of the decomposition on the cone T_k and its polar N_k.

Since $\|w\| \leq 1$, we can use (5.19) to bound the first term on the right-hand side, independently of w.

As for the second term, we have

$$[w]_{N_k}^\top \, [-\nabla f(x_k)]_{N_k} \quad \leq \quad \left\| [-\nabla f(x_k)]_{N_k} \right\| \, \left\| [w]_{N_k} \right\|.$$

Now, by definition of N_k, the projection of w onto this cone can be written as

$$[w]_{N_k} = \sum_{i \notin I_k^+} \beta_i^+ \, e_i - \sum_{j \notin I_k^-} \beta_j^- \, e_j,$$

where β_i^+ and β_j^- are nonnegative coefficients.

Since $x_k + w$ is feasible, we know that none of the components of w can exceed α_k. Therefore, we necessarily have $\| \, [w]_{N_k} \, \| \leq \sqrt{n} \, \alpha_k$. By definition of the projection onto the cones N_k, we also have

$$\| \, [-\nabla f(x_k)]_{N_k} \, \| \leq \| - \nabla f(x_k) \| \leq B_g,$$

Therefore, we can obtain a bound on the second term that does not depend on w. Together with the first one, this yields (5.20). □

Lemma 5.4 is thus the equivalent of Lemma 3.2 for the bound-constrained setting. We can thus apply the same analysis as in Chapter 3 to establish convergence. Indeed, Algorithm 5.1 relies on the same decrease condition and step size updates as Algorithm 3.1; we can thus retain the result of Lemma 3.1, which implies that the step size goes to zero for all realizations of the method. Provided we can guarantee satisfaction of the κ-feasible descent property sufficiently often, we can define an appropriate submartingale to ensure that this yields convergence of the framework. Theorem 5.1 states the result.

Theorem 5.1 *Let Assumptions 2.1, 2.2, 2.4 and 5.1 hold. Suppose that Algorithm 5.1 is applied using a polling set sequence $\{\mathfrak{D}_k\}$ of unit vectors that is (p, κ)-feasible descent with $p \geq p_0$, where $p_0 = \ln \theta / (\ln (\gamma^{-1} \, \theta)$. Then*

$$\mathbb{P} \left(\liminf_{k \to \infty} \chi(X_k) = 0 \right) \; = \; 1. \tag{5.21}$$

Following the lines of Section 2.3, we can also derive worst-case probabilistic complexity bounds on the number of iterations and evaluations necessary to satisfy

$$\inf_{0\leq l\leq k} \chi(x_l) \;<\; \epsilon, \tag{5.22}$$

for a given threshold $\epsilon \in (0,1)$. The resulting estimate is given below.

Theorem 5.2 *Let Assumptions of Theorem 5.1 hold, and suppose further that the sequence $\{\mathfrak{D}_k\}$ is (p,κ)-feasible descent with $p > p_0$. In addition, suppose that the forcing function is $\rho(\alpha) = c\,\alpha^2/2$ and that*

$$\epsilon \;\leq\; \frac{\mathcal{C}\,\alpha_0}{2\gamma}, \tag{5.23}$$

with $\mathcal{C} = \left((c+\nu_g)\,\kappa^{-1} + 2\sqrt{n}\right) B_g$. Then, the first index K_ϵ for which (5.22) holds satisfies

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{\beta\,\mathcal{C}^2}{c\,(p-p_0)}\epsilon^{-2}\right\rceil\right) \;\geq\; 1 - \exp\left[-\frac{\beta(p-p_0)\,\mathcal{C}^2}{8cp}\epsilon^{-2}\right]. \tag{5.24}$$

where β is a bound on $\sum_{k=0}^{\infty}\alpha_k^2$.

We now comment on the values taken by this bound in our two strategies. Suppose first that the polling sets are randomly generated following the technique described in Section 5.2.2, i.e. by sampling s directions in the generators of the approximate tangent cone. We know that by choosing $s > \lceil 2\,n\,p_0\rceil$ directions, the sequence will be (p,κ)-feasible descent with $p > p_0$ and $\kappa = 1/\sqrt{n}$; as a result, (5.24) above becomes

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{\beta\,(c+\nu_g+2)^2}{c\,(p-p_0)} n\,\epsilon^{-2}\right\rceil\right) \;\geq\; 1 - \exp\left[-\frac{\beta(p-p_0)\,(c+\nu_g+2)^2}{8\,c\,p} n\,\epsilon^{-2}\right]. \tag{5.25}$$

The previous bound thus yields a probabilistic worst-case estimate of the number of function evaluations in $\mathcal{O}\left(s\,\frac{n\epsilon^{-2}}{p-p_0}\right)$, holding with overwhelming probability. Note that we kept the term involving the probability p, since it can now depend on the dimension n (note however that the other constants remain independent on n and s). Assuming for instance $p_0 = 1/2$, we have that $p - p_0 = \mathcal{O}\left(\frac{1}{n}\right)$ and $s = n+1$, thus the bound is in $\mathcal{O}\left(n^3\,\epsilon^{-2}\right)$. This is one order of n worse than what one can obtain using $D_\oplus$ to solve bound-constrained or unconstrained problems Such a result should not come as a surprise, since we restricted ourselves to the use of directions from the set $D_\oplus$.

If we aim at exploiting subspace information in the spirit of Section 5.2.3, we observe a deterioration of the bound as κ is now of order of $\frac{1}{n}$ instead of $\frac{1}{\sqrt{n}}$. Even though less directions can be used in practice with that technique, in the worst case, it might be that subspaces are not worth exploiting and one should instead rely on coordinate directions, as in the previous strategy. In such a case, the bound deteriorates by an additional order of n, i.e., $\mathcal{O}\left(n^4\,\epsilon^{-2}\right)$.

Note that when there are only $n_b < n$ variables subject to bound constraints, a subspace of dimension $n - n_b$ will always be included in the approximate tangent cone. Therefore, if we let $\gamma = 2$ and $\theta = 0.5$, the worst-case number of evaluations decreases to $s = n_b + 2$ as it suffices to generate 2 directions in the subspace to have descent in probability. This also means that we will have $p - p_0 = \mathcal{O}\left(\frac{1}{n_b}\right)$, which yields a bound in

$$\mathcal{O}\left(n_b(n_b+2)\, n^2 \epsilon^{-2}\right).$$

This probabilistic technique may then prove beneficial when few variables are bounded. This makes sense as the method would need less directions to conform to the geometry of the feasible set.

5.3 Probabilistic feasible descent for linear equality constraints

In this section, we describe a straightforward extension of the unconstrained setting to handle linear equality constraints on the problem variables, i.e. we consider

$$\begin{cases} \min & f(x) \\ s.t. & A\,x = b. \end{cases} \tag{5.26}$$

Here $A \in \mathbb{R}^{n_e \times n}$, with $n_e \leq n$, with the convention that $n_e = 0$ if there are no constraints. We make the following assumption on this matrix.

Assumption 5.2 *Either $n_e = 0$, in which case the matrix A is empty and we are in the unconstrained case, or $n_e \geq 1$ and the matrix A is of full row rank.*

The presence of linear equality constraints can be viewed as a reduction on the size of the problem. As a result, solving the problem might be done at a cheaper cost than in the unconstrained case, provided we can introduce the linear relationship provided by the constraints within our algorithm.

5.3.1 Subspace-based direction generation techniques

In the presence of linear equality constraints, designing a method based on feasible descent directions means that at the k-th iteration, one must consider evaluating the function along a vector $d \in \mathbb{R}^n$ satisfying

$$d^\top \nabla f(x_k) < 0 \quad \text{and} \quad A\,d = 0. \tag{5.27}$$

Let $Z \in \mathbb{R}^{n \times (n-n_e)}$ be a basis matrix for the null space of A, which we will assume to be orthonormal for simplicity. To obtain a feasible descent direction, one need only choose a direction $p \in \mathbb{R}^{n-n_e}$ such that

$$p^\top Z^\top g_k = g_k^\top (Z\,p) < 0. \tag{5.28}$$

The vector $Z^\top g_k$ is called the *reduced gradient of f at x_k*; its opposite plays the same role as the negative gradient in unconstrained optimization. Note that contrary to the bound-constrained case, the step size does not intervene in the above feasibility requirement. For this reason, our analysis will be closely related to the unconstrained case. Indeed, instead of directly generating directions in the variable space, we may produce vectors in $\mathbb{R}^{n-n_e}$ and then poll along the directions defined by $Z\,D_k$, which are in $\mathbb{R}^n$ and will be feasible directions. This was the approach proposed by Liu and Zhang [85] for pattern-search-type methods. Our proposal is slightly different, and consists in generating directions in $\mathbb{R}^n$ and to multiply them by $Z\,Z^\top$. Definition 5.3 describes the quality we expect from such a polling set when linear equality constraints are present.

Definition 5.3 *Consider a realization of Algorithm 5.1 applied to Problem* (5.26)*. Let Z be an orthonormal basis for the null space of A and consider a feasible point x_k such that $\left\|-Z^\top \nabla f(x)\right\| \neq 0$. A set $D \subset \mathbb{R}^n$ is said to be* κ-feasible descent *for problem* (5.26) *at x_k if*

$$\max_{\substack{d\in D \\ \|Z^\top d\|\neq 0}} \frac{d^\top Z \left[-Z^\top \nabla f(x_k)\right]}{\|Z^\top d\|\|-Z^\top \nabla f(x_k)\|} \;\geq\; \kappa, \tag{5.29}$$

where $\kappa \in (0,1)$.

For instance, by choosing $D = D_\oplus$, the property is satisfied as by definition, $[Z\ -Z]$ is a positive basis for the null space of A (note that $[Z\,Z^\top\ -Z\,Z^\top]$ is a Positive Spanning Set for this subspace).

We again emphasize that the basis Z and the set D are not necessarily related. This is a key element of our analysis since it allows to generate random directions while ensuring feasibility, as we will see later.

The next definition is the probabilistic counterpart of Definition (5.3).

Definition 5.4 *Under the same considerations as in Definition 5.3, consider a sequence of randomly, independently generated sets $\{\mathfrak{D}_k\}$, and $(\kappa, p) \in (0,1)^2$. Consider the event*

$$E_k \;=\; \{\ \mathfrak{D}_k \text{ is a } \kappa\text{-feasible descent set for (5.26) at } x_k\ \}. \tag{5.30}$$

The sequence $\{\mathfrak{D}_k\}$ is called a (p,κ)-feasible descent sequence *for problem* (5.26) *if*

$$\mathbb{P}(E_0) \;\geq\; p \quad \textit{and} \quad \forall\, k \geq 1,\ \mathbb{P}\left(E_k \mid \sigma\left(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right)\right) \;\geq\; p. \tag{5.31}$$

To produce such a sequence, one can use the approach of Section 5.2.2, consisting here in selecting a random subset of the columns of $[Z\,Z^\top\ -Z\,Z^\top]$. Another possibility is to perform random generation within the null space of A, which can be assimilated to both the technique of Section 5.2.3 and that presented in Section 3.5.

For simplicity of exposure, we will now rely on polling sets that fall into Assumption 5.3.

Assumption 5.3 *The polling sets used in Algorithm 5.1 are of the form $Z\,Z^\top\,\mathfrak{D}_k$, where $\mathfrak{D}_k$ is a set of vectors in $\mathbb{R}^n$ that are normalized so that*

$$\forall\, d \in D, \quad \left\| Z^\top d \right\| = 1.$$

If Z is an orthonormal basis, it implies that $\left\| Z\,Z^\top d \right\| = 1$.

The trial points at iteration k thus are of the form $x_k + \alpha_k\, Z\, Z^\top d$, which guarantee their feasibility. Note that this normalization has been adopted in our numerical experiments; its motivation can be understood through the following lemma and proposition.

Lemma 5.5 *Let d be a random vector following a standard normal distribution in $\mathbb{R}^n$, which we denote by*

$$d \;\sim\; \mathcal{N}\left(0_{\mathbb{R}^n}, I_n\right).$$

Let $Z \in \mathbb{R}^{n\times(n-n_e)}$ be an orthonormal basis of a subspace of $\mathbb{R}^n$, of dimension $n - n_e$. Then

$$Z^\top d \;\sim\; \mathcal{N}\left(0_{\mathbb{R}^{n-n_e}}, I_{n-n_e}\right),$$

that is $Z^\top d$ follows a standard normal distribution in $\mathbb{R}^n$, and consequently

$$\frac{Z^\top d}{\|Z^\top d\|} \;\sim\; \mathcal{U}_{\mathbb{S}^{n-n_e-1}},$$

i.e., $\frac{Z^\top d}{\|Z^\top d\|}$ is uniformly distributed in the unit sphere in $\mathbb{R}^{n-n_e}$.

As a result, one can apply the proof technique from the unconstrained case (see Corollary 3.4) to conclude on the number of directions needed to produce a feasible descent sequence.

Proposition 5.4 *Suppose that the polling set sequence in Algorithm 5.1 is of the form $\left\{ Z\,Z^\top\,\mathfrak{D}_k \right\}$, where $\mathfrak{D}_k$ contains a fixed number m of directions randomly independently generated following a standard normal distribution in $\mathbb{R}^n$. Then, if*

$$m \;>\; \log_2\left(1 - \frac{\ln\theta}{\ln\gamma}\right), \tag{5.32}$$

it exists $p \geq p_0$ and $\tau > 0$ determined by θ and γ such that the polling set sequence is $(p, \tau/\sqrt{n-n_e})$-feasible descent for problem (5.26).

5.3.2 Convergence and complexity study

As in Section 5.2.4, we highlight the core differences between this setting and the unconstrained one rather than repeating an almost identical analysis. In the case of linear equality constraints, the reasoning is even more similar.

Lemma 5.6 *Let Assumptions 2.1 and 5.3 hold. Consider a realization of Algorithm 5.1, and assume that $Z^\top g_k \neq 0$ with k being the index of an unsuccessful iteration; then, if D_k is a κ-feasible descent set at x_k,*

$$\left\| Z^\top g_k \right\| \leq \frac{1}{\kappa} \left(\frac{\nu_g}{2} \alpha_k + \frac{\rho(\alpha_k)}{\alpha_k} \right). \tag{5.33}$$

Theorem 5.3 *Let Asssumptions 2.1, 2.2, 2.4 and 5.2 hold. Consider an application of Algorithm 5.1 with a polling set sequence generated following Assumption 5.3; suppose further that the sequence $\{\mathfrak{D}_k\}$ is a (p, κ)-feasible descent sequence for problem (5.26), with $(\kappa, p) \in (0,1)^2$ and $p \geq p_0 = \ln(\theta)/\ln(\gamma^{-1}\theta)$.*

Then, the sequence of iterates produced by the algorithm satisfies

$$\mathbb{P}\left(\liminf_{k \to \infty} \|Z^\top G_k\| = 0 \right) = 1. \tag{5.34}$$

The worst-case probabilistic complexity bounds of interest concern here the number of iterations and evaluations necessary to satisfy

$$\inf_{0 \leq l \leq k} \left\| Z^\top \nabla f(x_l) \right\| < \epsilon, \tag{5.35}$$

with $\epsilon \in (0,1)$.

Theorem 5.4 *Let Assumptions of Theorem 5.3 hold, and suppose that the sequence $\{\mathfrak{D}_k\}$ is (p, κ)-feasible descent with $p > p_0$. Additionally, suppose that $\rho(\alpha) = c\,\alpha^2/2$ and*

$$\epsilon \leq \frac{(c + \nu_g)\,\alpha_0}{2\kappa\gamma}, \tag{5.36}$$

the first index K_ϵ for which (5.35) does not hold satisfies

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{\beta\,(c+\nu_g)^2}{c(p-p_0)\kappa^2} \epsilon^{-2} \right\rceil \right) \geq 1 - \exp\left[-\frac{\beta(p-p_0)\,(c+\nu_g)^2}{8cp\kappa^2} \epsilon^{-2} \right] \tag{5.37}$$

where β is a bound on $\sum_{k=0}^{\infty} \alpha_k^2$.

The results of Theorem 5.2 are more insightful than in the bound-constrained case. Indeed, a similar study as the one proposed in the unconstrained case establishes that in the subspace of dimension $n - n_e$ spanned by Z, it is possible to randomly generate directions so that the corresponding polling set sequence is $(p, \tau/\sqrt{n-n_e})$-descent, with p and τ independent of n and n_e. The resulting bound on the number of iterations is thus in

$$\mathcal{O}\left((n - n_e)\,\epsilon^{-2} \right), \tag{5.38}$$

while, if we let m be the (maximum) number of vectors contained in the polling sets at every iteration, the bound on the number of function evaluations is

$$\mathcal{O}\left(m\,(n - n_e)\,\epsilon^{-2} \right), \tag{5.39}$$

both results holding with overwhelming probability as $\epsilon \to 0$.

Since s can be taken independently of n_e and n, the resulting complexity bound is of order $\mathcal{O}((n - n_e)\,\epsilon^{-2})$. This bound not only represents an improvement compared to the deterministic case, where one would consider a positive spanning set for the null space of A, but also brings new insights on the interest of maintaining equality linear constraints in an optimization problem. Indeed, even if those constraints cannot be trivially solved so as to reduce the initial problem to one with an equivalent solution (in $\mathbb{R}^{n-n_e}$), one may still aim at solving the original problem as efficiently as it would be solved in $\mathbb{R}^{n-n_e}$. In that case, the results from Chapter 3 indicate that the complexity of a direct-search method based on probabilistic descent would be in $\mathcal{O}\left((n - n_e)\,\epsilon^{-2}\right)$, which is what we obtain in the present chapter. The effect of randomness in the linearly equality-constrained case thus seems the same than in the unconstrained one.

5.3.3 Addressing linear inequality constraints

An extension of the strategy proposed in the previous section can be considered through an active-set strategy, such as the ones proposed in [75], so as to handle linear inequality constraints by considering those close to the current point as linear equalities. The main drawback of this approach is that one must compute as much bases Z as there are possibilities for the tangent cone. Since the number of inequality constraints can be large, this may result in significant computation time to determine those bases, even though they would be determined prior to the execution of the method (see [77] for a description of classical techniques). Besides, in the inequality-constrained case, degeneracy of the constraints at a particular point is often encountered, particularly whenever the linear inequality constraints are considered under a general form that may include bounds [5].

5.4 Numerical results

In this section, we present numerical results for problems coming from the `CUTEst` collection [65]. We implemented our various strategies in MATLAB and use the built-in `patternsearch` function [89] for comparison. We set the options of this function and ours so that they would follow the Generating Set Search approach for bound-constrained or linearly-constrained optimization problems given in [75], our objective being to remain as close as the default settings as possible. More specifically, for all methods, the budget was taken to be $2000\,n$ evaluations, and the minimum step size allowed was given by $10^{-6}\,\alpha_0$, with $\alpha_0 = 1$. The increase and decrease parameters for the step size were set at $\gamma = 2$ and $\theta = 0.5$ (which means that $p_0 = 1/2$).

We ran the methods with these budget and limit sizes, yielding a best obtained value f_{best} for all problems considered. Performance profiles [50, 95] were then established for a given tolerance ϵ by computing the number of function evaluations needed by each method to reach an iterate x_k at which

$$f(x_k) - f_{best} \;<\; \epsilon\,(f(x_0) - f_{best})\,.$$

Such profiles plot the fraction of problems for which the method is the least expensive in terms of function calls (y-axis), then the fraction of problems solved given an additional expense with respect to the best one. Note that methods based on random directions were run ten times and the mean of the results was used.

5.4.1 Bound-constrained problems

We first present results on problems that only enforce bound constraints on their variables. For each of these problems, we chose small sizes so that the benchmark was composed of 63 problems, with 29 being of dimension at least 10.

Three variants of our randomized setting are considered in those results. The variant 0 simply consists in choosing a random permutation of the columns of the coordinate set: it appeared relevant to include this variant in our algorithm to verify that the differences between our strategies and the deterministic ones were due to more than a random ordering of the polling directions (note that it clearly relies on a feasible descent set). The variant 1 corresponds to the sampling approach described in Section 5.2.2 for bound constraints, with a sample size adjusted at every iteration to the minimal number of vectors needed to be feasible descent with sufficient probability. Finally, the variant 2 is related to the subspace approach described in Section 5.2.3.

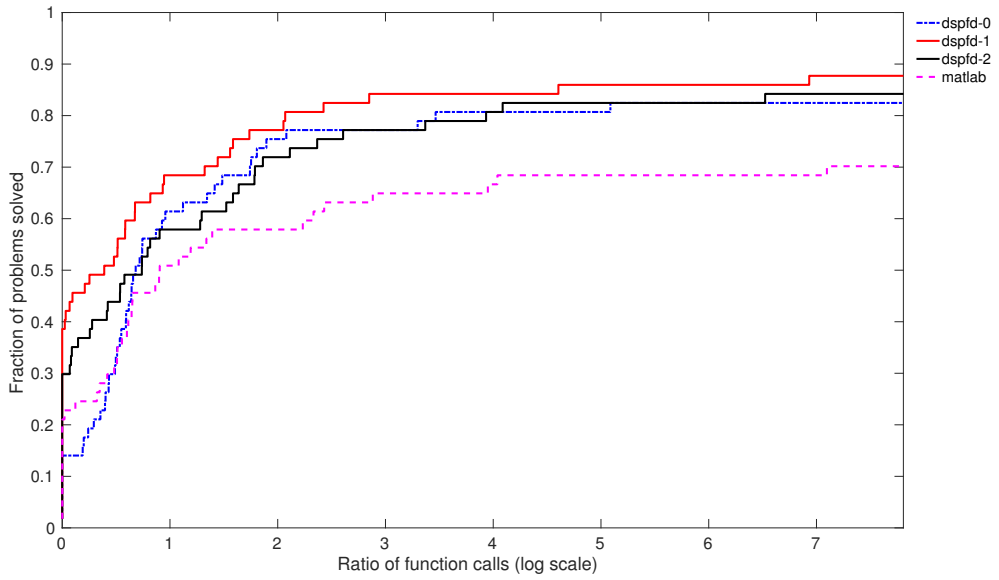
As expected, the coordinate set performs quite well in this setting; yet, with a sample size of order of half of the directions within the set of free bound constraints, we are able to reach similar (often improved) final function values, with a considerably lower number of function evaluations.

Because of this economy in the polling process, the probabilistic techniques exhibit superior performance compared to the MATLAB implementation, see Figure 5.1. The technique 1, described in Section 5.2.2, outperforms the others. Rather than being tied to partial separability of the test problems (applying a rotation to the variables prior to function evaluation essentially yields the same profiles), this performance can be explained by the intrinsic properties of $D_{\oplus}$ with respect to the bound constraints.

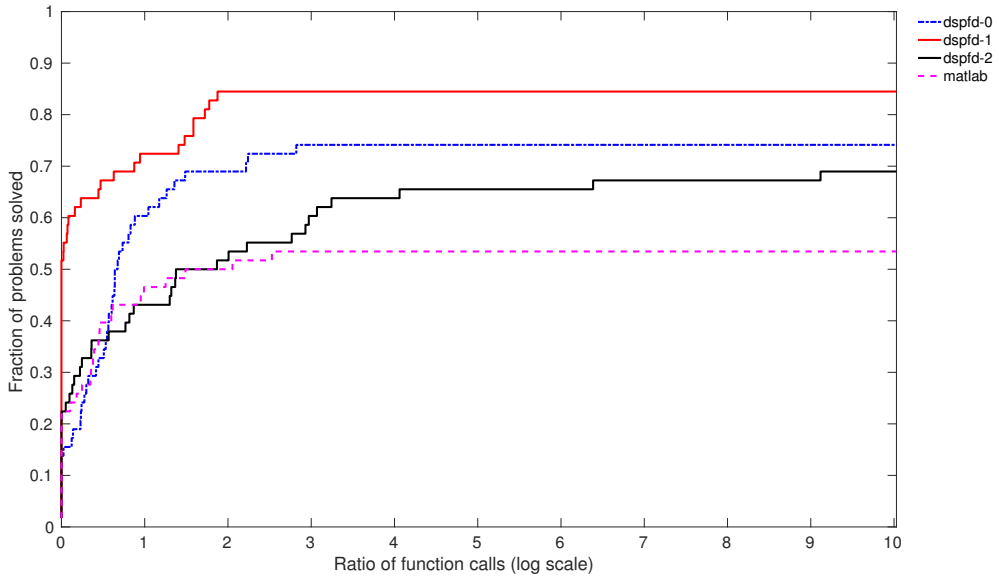
5.4.2 Linearly-constrained problems

We first tested the strategies described in Section 5.3 on ten tests problems from the `CUTEst` collection for which only linear equality constraints were present. Here the variants 0 and 1 consisted in using $[Z\,Z^\top - Z\,Z^\top]$ as search directions, respectively with a random permutation and a random subset selection. In the variant 2, we drew two opposite directions within the null space of A, so as to encounter one of descent type with a probability higher than $p_0 = 0.5$.

Table 5.1 presents the results obtained on those problems by running the methods until either α_k decreased below $10^{-6}\,\alpha_0$ or the budget of $2000\,n$ function evaluations was exceeded. On such examples, our variants of Algorithm 5.1 managed to reach comparable (even better) function values compared to the MATLAB function in significantly less function evaluations. Interestingly, when all methods reached the same final value, the `dspfd-2` was the least expensive in doing so.



(a) $\epsilon = 10^{-3}$.



(b) $\epsilon = 10^{-6}$.

Figure 5.1: Performance of three variants of Algorithm 5.1 versus MATLAB `patternsearch` on bound-constrained problems.

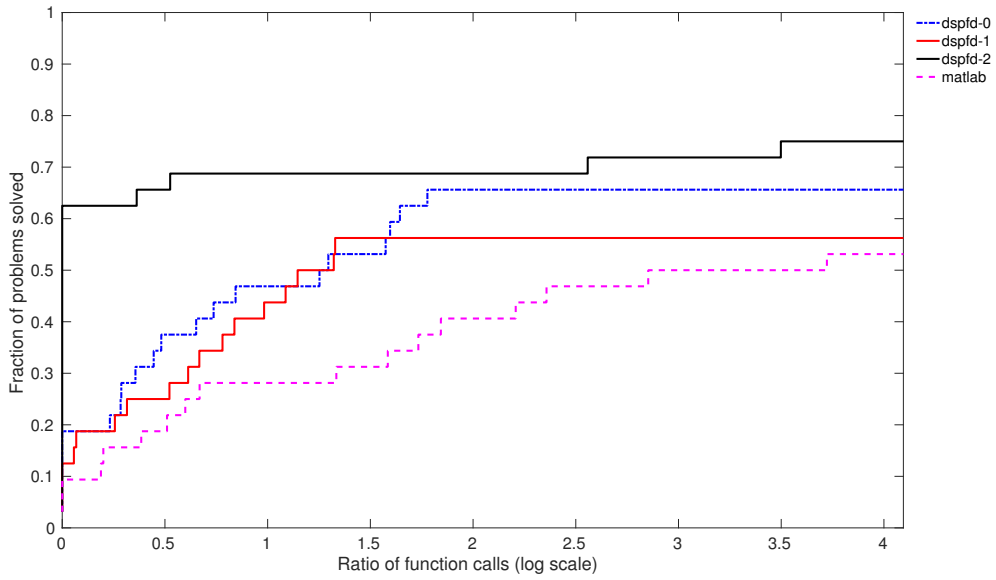
Table 5.1: Number of evaluations and final value on problems with linear equality constraints.

Problem	Dim	Lin. Eq.	dspfd-0	dspfd-1	dspfd-2	matlab
BT3	5	3	201 *1e+39*	101 *1e+39*	41 *1e+39*	201 *1e+39*
HIMMELBA	2	2	81 *0.0*	41 *0.0*	41 *0.0*	81 *0.0*
HS9	2	1	197 *-0.5*	69 *-0.5*	52 *-0.5*	229 *-0.5*
HS28	3	1	249 *4e-31*	176 *2e-31*	157 *8e-14*	182 *1e-30*
HS48	5	2	354 *1e-30*	203 *8e-31*	211 *2e-13*	977 *6e-13*
HS49	5	2	10000 *1e-06*	9025 *7e-10*	9476 *3e-07*	10000 *1e-07*
HS50	5	3	438 *3e-26*	290 *4e-27*	185 *5e-13*	448 *1e-26*
HS51	5	3	281 *7e-31*	152 *9e-31*	144 *3e-14*	322 *8e-30*
HS52	5	3	201 *1e+40*	101 *1e+40*	41 *1e+40*	201 *1e+40*
ZANGWIL3	3	3	121 *0.0*	61 *0.0*	41 *0.0*	121 *0.0*

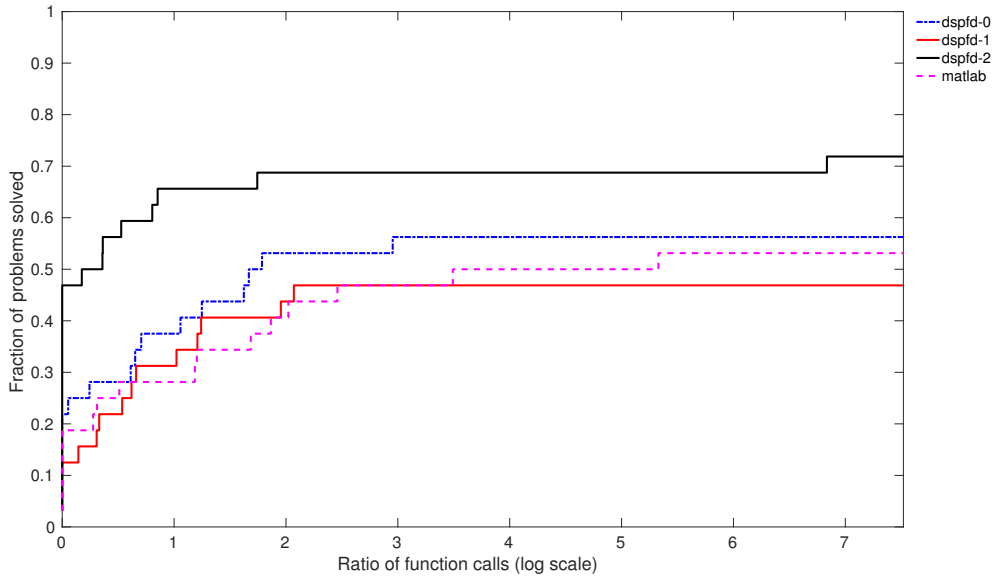
Our final experiment is made on a benchmark of problems that includes the previous ten problems, as well as 34 problems with both linear equality constraints and bounds on the variables. This benchmark thus consists of 44 `CUTEst` problems, with dimension varying between 2 and 32.

Note that feasibility is a more critical issue for general problems involving the two types of constraints. Indeed, the basis of the null space of A is generally not aligned with the coordinate axes: thus, feasibility of the directions forming the basis is no longer guaranteed, nor can it be certified that the directions are able to produce feasible descent. Still, it appeared interesting to test our method in such an environment. The polling strategies remained the same as for the case of sole linear equality constraints, but we only considered directions in $\begin{bmatrix} Z\,Z^\top & -Z\,Z^\top \end{bmatrix}$ for which a displacement of α_k was possible without violating the bounds. We then proceeded as in Section 5.2, only with the columns of $Z\,Z^\top$ playing the role of the coordinate directions.

Figure 5.2 presents the resulting performance profiles, obtained by aiming at a convergence criterion. Note that on those problems, the performances of the variants 0 and 1 are much closer to that of MATLAB's `patternsearch`, and the potential improvement they might generate seems rather due to random ordering of the vectors than to the sampling strategy. However, the variant `dspfd-2`, that has more freedom in the choice of its directions by randomly generating directions in subspaces, is by far the most efficient method. This supports our claim that identifying subspaces in which one can proceed as if the problem was unconstrained is beneficial for our probabilistic strategies.



(a) $\epsilon = 10^{-3}$.



(b) $\epsilon = 10^{-6}$.

Figure 5.2: Performance of three variants of Algorithm 5.1 versus MATLAB `patternsearch` on problems with linear equality constraints, with or without bounds.

5.5 Conclusion of Chapter 5

We presented several probabilistic techniques for solving bound-constrained and linearly-constrained problems with direct-search methods that do not rely on generating sets at every iteration. As enlightened by our analysis, the cost of such techniques is essentially influenced by the number of constraints. Linear equality constraints produce reduction on the dimensionality of the problem, and, consequently, asymptotically require a lower number of function evaluations to reach a given tolerance. On the contrary, bound constraints tend to have the opposite effect in that they may need to use a significant amount of coordinate directions to guarantee feasible descent. This brings supplementary insights regarding the relevancy of the coordinate set for direct-search methods applied to bound-constrained problems. The probabilistic reasoning shows that using only a sample of those vectors can be more economical and potentially improve the performance, while converging with probability one to a stationary point of the problem.

Our study provided several possibilities for using fewer directions while maintaining convergence guarantees when one type of constraints is to be addressed, yet our strategies appear to be of use even in general formulations. Indeed, our variants based on probabilistic properties were observed to outperform the standard direct-search instance available in the MATLAB toolbox, on a benchmark including problems for which the optimization variables were subject to bounds and/or linear equality constraints. Such results strongly encourage to use probabilistic approaches to address linearly-constrained problems, especially in the context of costly function evaluations. Exploiting (feasible) subspaces stands out as a key feature that enables to reduce the number of calls to the objective, thanks to a random generation argument already employed in the unconstrained setting.

Chapter 6

Second-order results for deterministic direct search

When applied to smooth problems, derivative-free algorithms mostly focus on approximating the information brought by the first-order derivative. Although there exist frameworks, essentially of model-based type, that care to exploit second-order information whenever present, such approaches are often associated with a cost that is prohibitive in practice. This is particularly true when the methods are built so as to ensure convergence towards a second-order stationary point. Still, the potential improvement brought by second-order considerations may be worth the expense, in that it can improve the best attainable value for a given budget of function calls, as well as the general robustness of the algorithm.

This chapter presents and complements the existing results regarding second-order properties of the direct-search class of methods. Section 6.1 introduces the challenges related to second-order in the DFO setting. In Section 6.2, the analysis of an elementary direct-search algorithm is performed from a second-order viewpoint, showing both the potential and the limitations of this basic framework. Section 6.3 presents an algorithm that is provably second-order convergent, for which complexity bounds can be obtained. The numerical behavior of the new method is investigated in Section 6.4, where second-order considerations prove themselves beneficial on nonconvex problems.

6.1 Exploiting negative curvature in a derivative-free environment

Second-order aspects have been shown to be beneficial in the derivative-based setting, even more when the function to optimize is not convex [63, 94]. Indeed, in that case, the Hessian matrix can have negative eigenvalues, and decrease in the function value might be possible along the associated eigenvectors. The goal of a *second-order convergent* optimization method is to compute both descent directions and *negative curvature* directions, so as to ensure convergence of the sequence of iterates towards a second-order stationary point (at which the gradient is zero and the Hessian is positive semidefinite).

Most derivative-free schemes are only concerned with approximating the first-order derivative, and therefore do not assume existence of higher-order information. Even though theoretical analysis has been proposed for second-order convergent methods (mostly inspired by the derivative-based literature), such aspects are often discarded in practical implementations, due to their unavoidable cost.

In the model-based family of algorithms, it is possible to design second-order globally convergent derivative-free methods, provided the associated models satisfy suitable approximation properties [37]. One way to fall into such assumptions consists in building polynomial approximation models using the values of $\frac{(n+1)(n+2)}{2}$ points around the current iterate.

For the direct-search class, intrinsic second-order properties of the frameworks have been studied in the cases of pattern search [2], mesh adaptive direct search [3] and generating set search [6], the third proposal following an algorithm presented by Frimannslund and Steihaug [57] based on approximating the Hessian matrix and using the resulting eigenvectors as polling directions. Note that in all these cases, the computation of negative curvature directions is not the goal that is pursued. Consequently, the methods may not possess global second-order convergence guarantees, and none of them was endowed with a complexity analysis.

6.2 Weak second-order criticality measure and associated results

In this section, we provide a study of the basic direct-search scheme of Algorithm 2.1 in a second-order perspective. Similarly to what was done in [2, 3], we want to understand if second-order information (provided it exists) can be gathered by an elementary direct-search method.

6.2.1 Second-order in a general direct-search framework

In order to take second-order aspects into account, we need to strengthen the usual assumptions for first-order analysis. Assumption 6.1 relates to the regularity of the objective, while Assumption 6.2 corresponds to the forcing function.

Assumption 6.1 *The function f is twice continuously differentiable with Lipschitz-continuous gradient and Hessian, of respective Lipschitz constants ν_g and ν_H.*

Assumption 6.2 *The forcing function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ satisfies the following properties:*

i) ρ is non-decreasing,

ii) $\rho(\alpha) = o(\alpha^2)$ when $\alpha \to 0^+$.

Note that when $t \to 0^+$, one has $\rho(\alpha) = o(\alpha)$, so a function satisfying Assumption 6.2 also verifies the first-order requirement of Assumption 2.4.

For simplicity of exposure, we also make the following hypothesis.

Assumption 6.3 *The Search Step of Algorithm 2.1 is omitted or, equivalently, the set of search directions is empty at every iteration.*

We finally look at the suitable properties that are to be satisfied by the polling sets. As described in Chapter 2, the use of PSSs is a common way of ensuring that a deterministic direct-search method considers descent directions. In order to maintain the first-order properties of the framework, we will thus assume that each polling set contains a PSS.

In addition, existing second-order convergence analyses are based on *symmetric* polling sets for a refining subsequence. Such symmetric sets have the property that $-D = \{-d \mid d \in D\} = D$, which allows to take advantage of evaluations of the function in opposite directions. This is fundamental if one aims at considering the second-order difference scheme (for a Lipschitz continuous Hessian)

$$f(x_k + \alpha_k\, d_k) - 2\, f(x_k) + f(x_k - \alpha_k\, d_k) \;=\; \alpha_k^2\, d_k^\top\, \nabla^2 f(x_k)\, d_k + \mathcal{O}(\alpha_k^3) \tag{6.1}$$

that serves as an approximation of the second-order directional derivative. This being said, one can still prove results in the more general case where each polling set admits a symmetric subset. This case is described in the following assumption.

Assumption 6.4 *The polling sets D_k are finite Positive Spanning Sets of unitary vectors, such that the* symmetric part of D_k *defined by*

$$V_k \;=\; \{d \in D_k \mid -d \in D_k\}$$

is not empty, and that the sequence of cosine measures is bounded from below by $\kappa_g > 0$.

A relevant example of such a sequence is $D_k = [Q \;\; -Q] = Q\, D_\oplus$, where Q is a rotation matrix. In that case, we have $D_k = V_k$ at each iteration.

Under the previous assumptions, we can still derive the proof of Lemma 2.1, thus resulting in the convergence of the step size sequence. We restate the property below for convenience.

Lemma 6.1 *Under Assumptions 2.2 and 6.2 to 6.4,* $\lim_{k\to\infty} \alpha_k = 0$.

Lemma 6.2 enlightens the second-order property that can be stated using V_k.

Lemma 6.2 *Under Assumptions 6.1 and 6.4, consider an iteration of index $k \in U$. One has*

$$\|\nabla f(x_k)\| \;\leq\; \kappa_g^{-1} \left(\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\, \alpha_k \right) \tag{6.2}$$

and

$$\min_{d \in V_k} d^\top \nabla^2 f(x_k)\, d \;\geq\; -\left(\frac{2\,\rho(\alpha_k)}{\alpha_k^2} + \frac{\nu_H}{3}\, \alpha_k \right). \tag{6.3}$$

Proof. The proof of (6.2) is identical to that of Lemma 2.2, therefore we focus on proving the second-order property (6.3).

Consider a direction $d \in V_k$. Given that both d and $-d$ do not fulfill the sufficient decrease condition, we can sum the two corresponding equations. We thus have

$$-2\,\rho(\alpha_k) \;\leq\; f(x_k + \alpha_k\, d) + f(x_k - \alpha_k\, d) - 2\, f(x_k).$$

Thus, a second-order Taylor expansion of both $f(x_k + \alpha_k\, d)$ and $f(x_k - \alpha_k\, d)$ leads to

$$-2\,\rho(\alpha_k) \;\leq\; \alpha_k^2\, d^\top \nabla^2 f(x_k)\, d + \frac{\nu_H}{3}\, \alpha_k^3, \tag{6.4}$$

thanks to the Lipschitz continuity of $\nabla^2 f$. In particular, (6.4) holds for the couple of directions that realize the minimum of $d^\top \nabla^2 f(x_k) d$ in V_k and the relation (6.3) can be easily derived. □

The previous result indicates that we can define a directional measure of second-order optimality that will play a similar role as the cosine measure in first-order optimality proofs. This measure was introduced in [66] as the *Rayleigh measure* of a set of vectors, expressed for a given matrix. This definition draws its inspiration from both the cosine measure of a set of vectors at a particular vector (see Definition 2.3), and the Rayleigh quotient, a useful tool while dealing with second-order optimality [18]. We remark that a similar quantity was used by Gratton et al. [68] in the case of trust-region methods, although no specific terminology was introduced.

Definition 6.1 *Let D be a set of unitary vectors in $\mathbb{R}^n$ and A an n-by-n real symmetric matrix. The* Rayleigh measure *of D with respect to A is given by:*

$$\mathrm{rm}\,(D, A) \;=\; \min_{d \in D}\; d^\top\, A\, d. \tag{6.5}$$

This measure approximates the lowest eigenvalue of A, the minimum value of the Rayleigh quotient, by a discrete minimum Rayleigh quotient among all vectors in D. This approximation is an exact one when D contains an eigenvector associated to the minimum eigenvalue of A, in which case the Rayleigh measure is equal to this lowest eigenvalue. Moreover, if A has at least one negative eigenvalue, the sign of the Rayleigh measure provides information about the directions in D that correspond to negative

values of the Rayleigh quotient. Such values indicate that the corresponding direction and its opposite are negative curvature directions. Besides, the Rayleigh measure of V_k with respect to the Hessian matrix appears in (6.3). This naturally encourages the use of this measure as a substitute to the minimum Hessian eigenvalue, which is the usual second-order criterion in derivative-based optimization.

6.2.2 Weak second-order global convergence results

We now establish a second-order property related to the Rayleigh measure, hence to the partial curvature information we are able to collect at each iteration. This property generalizes the *pseudo-second order optimality* conditions presented for pattern search [2], and was called *weak second-order optimality* in [68] where the authors consider a trust-region framework with incomplete curvature information. We will use the latter terminology and propose a weak second-order optimality criterion based on the sequence of Rayleigh measures. Note that such a technique does not guarantee that the algorithm is able to avoid converging to a maximizer or a saddle point. Abramson showed in [2] that Algorithm 2.1 may converge to critical points where the Hessian matrix has zero eigenvalues, even though those points are not local minimizers. However, for these examples, one still obtains the property called *weak second-order optimality* by Gratton et al. [68], i.e., second-order optimality guarantees with respect to a set of directions. Theorem 6.1 presents an even more general formulation.

Theorem 6.1 *Suppose that Assumptions 6.1 to 6.4 hold. Then*

$$\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\operatorname{rm}\left(V_k, \nabla^2 f(x_k)\right)\right\} \;=\; 0. \tag{6.6}$$

In the specific case where $D_k = V_k$, the result becomes:

$$\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\operatorname{rm}\left(D_k, \nabla^2 f(x_k)\right)\right\} \;=\; 0. \tag{6.7}$$

Proof. From Lemma 6.2, we know that for any unsuccessful iteration of index k,

$$\|\nabla f(x_k)\| \;\leq\; \frac{1}{\kappa_g}\left[\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k\right] \tag{6.8}$$

and

$$-\operatorname{rm}\left(V_k, \nabla^2 f(x_k)\right) \;\leq\; \frac{2\rho(\alpha_k)}{\alpha_k^2} + \frac{\nu_H}{3}\,\alpha_k \tag{6.9}$$

hold, hence

$$\max\left\{\|\nabla f(x_k)\|, -\operatorname{rm}\left(V_k, \nabla^2 f(x_k)\right)\right\} \;\leq\; \max\left\{\frac{1}{\kappa_g}\left[\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k\right], \frac{2\rho(\alpha_k)}{\alpha_k^2} + \frac{\nu_H}{3}\,\alpha_k\right\}. \tag{6.10}$$

Lemma 6.1 ensures that there exists an infinite subsequence of unsuccessful iterations. For such a subsequence, both (6.8) and (6.9) hold, and the right part of each inequality

goes to zero when k goes to infinity thanks to Assumption 6.2. We thus conclude that (6.6) holds. The specific case where $D_k = V_k$ is immediate. □

Our analysis shows that any direct-search method that follows the framework of Algorithm 2.1 and imposes a sufficient decrease of $o(\alpha^2)$ exhibits weak second-order properties. In practice, if we were to use the same symmetric set of directions at each iteration, we would know that, at any limit point, the corresponding Rayleigh measure is nonnegative. This result is tight in the sense that additional properties on the directions are needed to ensure that the method does not converge to a first-order stationary point that is not a minimum. Consider, for instance, applying Algorithm 2.1 to the following function presented in [6]:

$$f_1(x, y) \;=\; (9\,x - y)\,(11\,x - y) + \frac{x^4}{2}, \tag{6.11}$$

with $x_0 = (0, 0)^\top$ as the initial point and $D_k = [e_1 \quad e_2 \quad -e_1 \quad -e_2]$ for all k. One sees that the method cannot move away from the origin, which is a saddle point. In that case, the coordinate directions and their opposites are not of negative curvature, as the Rayleigh measure is equal to zero at each iteration; the method is thus weakly second-order convergent on this function, but not second-order globally convergent.

The following corollary clarifies the link between (6.6) and the existing second-order results based on limit of refining directions [2, 3, 6].

Corollary 6.1 *Under the assumptions of Theorem 6.1, suppose that the sequence of iterates $\{x_k\}$ is bounded. Then there exists a subsequence of iterates $\{x_k\}_{k\in K}$ converging to a limit point x_* such that $\nabla f(x_*) = 0$.*

Define the set of refining directions V_* by

$$V_* \;=\; \{d \in \mathbb{R}^n \mid \exists L \subset K, \{d_l\}_{l\in L} \to d, \forall\, l \in L, d_l \in D_l\}\,.$$

Then the curvature at x_* is nonnegative along the directions in V_*, i.e.,

$$\forall\, d \in V_*,\; d^\top \nabla^2 f(x_*)\, d \;\geq\; 0. \tag{6.12}$$

Note that if V_* is dense in the unit sphere, the limit point x_* is a second-order critical one; the second-order optimality is thus assured by a similar argument as for the MADS methods [3], although those algorithms do not enforce sufficient decrease.

PSSs without symmetric parts When one aims for first-order convergence results using PSSs, it is not mandatory for those PSSs to have a non-empty symmetric part [75]. One might thus wonder if second-order results are still provable using PSSs for which the symmetric part is empty.

Given a PSS $D = [d_1 \cdots d_m]$, there always exist m nonnegative scalars $(\beta_i)_{i=1,m}$ such that $\sum_{i=1}^m \beta_i\, d_i = 0$ [43]. Considering Algorithm 2.1 and using these scalars, one can proceed as in Lemma 6.2 to arrive at the following relation:

$$\sum_{i=1}^{m} \beta_i\, f(x_k + \alpha_k\, d_i) - \left(\sum_{i=1}^{m} \beta_i\right) f(x_k) \;\geq\; -\left(\sum_{i=1}^{m} \beta_i\right) \rho(\alpha_k), \tag{6.13}$$

which leads to

$$\sum_{i=1}^{m} \beta_i \, d_i^\top \nabla^2 f(x_k) d_i \;\geq\; -\sum_{i=1}^{m} \beta_i \left(\frac{\rho(\alpha_k)}{\alpha_k^2} + \frac{\nu_H}{6} \, \alpha_k \right). \tag{6.14}$$

We may then derive the analysis in a similar way as before, and obtain a result on the convergence of a weighted sum of Rayleigh quotients under the appropriate assumptions, that is:

$$\limsup_{k \to \infty} \sum_{i=1}^{|D_k|} \beta_i^{(k)} \, (d_i^{(k)})^\top \, \nabla^2 f(x_k) \, d_i^{(k)} \;\geq\; 0, \tag{6.15}$$

where, for all k, $\sum_i \beta_i^{(k)} \, d_i^{(k)} = 0$.

One notices that we obtain a weaker result than in the symmetric case. Indeed, the combination $\{\beta_i^{(k)}\}_i$ depends on the direction set, which possibly changes at each iteration, and the meaning of (6.15) is then unclear. When the sets are symmetric, however, we can exploit this symmetry by constructing $|D_k|/2$ nonnegative combinations such that only the coefficients corresponding to a couple of opposite directions are not equal to zero (as we have seen in the proof of Lemma 6.2). The resulting properties are stronger as they involve the Rayleigh measure.

6.3 A provably second-order globally convergent direct-search method

The goal of this section is to improve the second-order results of Section 6.2 in order to obtain a method that is second-order globally convergent in the usual sense. Ideally, we would like to define a second-order property on the polling directions which would be equivalent to the positive spanning property for the first order. In derivative-based methods, this is done by assuming that one of the directions is of *negative curvature*. This means that if the Hessian $\nabla^2 f(x_k)$ has a minimum eigenvalue $\lambda_k < 0$, it exists $d \in D_k$ such that

$$d^\top \nabla f(x_k) \;\leq\; 0 \quad \text{and} \quad d^\top \nabla^2 f(x_k) \, d \;\leq\; \kappa \, \lambda_k, \tag{6.16}$$

with $\kappa \in (0, 1)$ independent of k. Such a requirement is classical in curvilinear line search methods [86, 94, 102] and second-order convergent line-search frameworks [63]. To generate such directions, one uses linear algebra techniques such as the Bunch-Parlett factorization of the Hessian, together with a Krylov subspace method [63].

In a derivative-free context, we do not have access to the Hessian matrix or its product with a vector, but we can estimate Rayleigh quotients. The first part of (6.16) is easy to satisfy through the use of symmetric polling sets, but the second inequality poses a harder problem. Indeed, it can be rewritten as follows:

$$\mathrm{rm}(D_k, \nabla^2 f(x_k)) \;\leq\; \kappa \; \mathrm{rm}\left(\mathbb{S}^{n-1}, \nabla^2 f(x_k) \right) \;=\; \kappa \left(\min_{d \in \mathbb{S}^{n-1}} d^\top \nabla^2 f(x_k) \, d \right) \;=\; \kappa \, \lambda_k, \tag{6.17}$$

where $\mathbb{S}^{n-1}$ denotes the unit sphere in $\mathbb{R}^n$. It thus appears that a direct-search method attempts to estimate at each iteration the solution of a quadratic problem with a finite number of directions, and given the lack of knowledge on the quadratic form itself, it seems rather demanding to ask for such an approximation.

In fact, derivative-free schemes that exploit curvature do not rely directly on vectors to capture these aspects, but rather on matrices that represent Hessian approximations [6, 37]. Although the resulting process does not necessarily ensure second-order convergence [16, 91], it is efficient in that it uses a matrix to estimate matricial information. This is why we will follow this approach to design our algorithm.

6.3.1 Using a Hessian approximation to determine additional directions

The instance of generating set search using curvature information presented in [57] is an important example of a second-order globally convergent method. This algorithm uses a Hessian approximation which is updated along the iterations (the update may not occur at every iteration). In the unconstrained case, the new directions are then obtained by computing the eigenvectors of the approximate Hessian. It is then the quality of the approximation, of the order of the step size, that leads to second-order global convergence.

This approach requires the use of PSSs of the form $[Q \ -Q]$, where Q is an orthogonal matrix. However, it is known [43] that both positive spanning sets and positive bases are not necessarily made of $2\,n$ vectors, nor are they necessarily symmetric. We thus would like to extend the idea of [57] in a more general setting. In addition, we would like a method that does not completely overtakes the framework of Algorithm 2.1; the expense of searching for negative curvature should not intervene unless the usual first-order approach has failed. This is the second objective of our algorithm. Last but not least, the amount of function evaluations at each iteration should be of order of n^2, to be in accordance with the methods that build a Hessian approximation or use models with second-order accuracy.

The above requirements lead to the direct-search instance described by Algorithm 6.1 and called AHDS, for Approximate Hessian-based Direct Search. It is close in spirit to the superlinearly convergent method developed by Mifflin [91], although we do not compute a gradient approximation. Here, we rather focus on exploiting negative curvature if possible.

Note that in case of a successful iteration with a direction in D_k, the method behaves like Algorithm 2.1 with a PSS. Remark also that we always require a decrease using $\rho(\alpha_k)$ whether the directions are unitary or not, but this does not affect the convergence nor the complexity analyses, which relate only to the unitary polling directions.

6.3.2 Second-order global convergence of the new method

Having presented our algorithm, we now show that it is indeed second-order globally convergent. The proof requires two intermediate results, that respectively emphasize

Algorithm 6.1: Approximate Hessian-based Direct-Search algorithm (AHDS)

Choose an initial point $x_0 \in \mathbb{R}^n$, as well as $0 < \theta < 1 \leq \gamma$, $0 < \alpha_0 < \alpha_{\max}$.
Define a *forcing function* $\rho : \mathbb{R}^+ \to \mathbb{R}^+$.
for $k = 0, 1, 2, \ldots$ **do**

1 - Poll Step

(a) Generate a Positive Spanning Set D_k of unitary vectors. If there exists $d \in D_k$ such that

$$f(x_k + \alpha_k\, d) \;<\; f(x_k) - \rho(\alpha_k), \tag{6.18}$$

then declare iteration k successful with $d_k = d$ and go to the update step.

(b) Otherwise, if there exists $d \in D_k$ such that (6.18) holds for -d, then declare the iteration successful with $d_k =$ -d and go to the update step.

(c) Otherwise, choose B_k as a subset of D_k with n linearly independent directions, which we index as $d_1, \cdots, d_n$.

(d) If there exists $d \in \{d_i + d_j, 1 \leq i < j \leq n\}$ such that (6.18) holds, then declare the iteration successful with $d_k = d$, and go to the update step.

(e) Otherwise, define the matrix H_k by setting for all $(i, j) \in \{1, \ldots, n\}^2, i < j$:

$$\begin{aligned} (H_k)_{ii} &= \tfrac{f(x_k+\alpha_k\, d_i)-2\, f(x_k)+f(x_k-\alpha_k\, d_i)}{\alpha_k^2}, \\ (H_k)_{ij} &= \tfrac{f(x_k+\alpha_k\, d_i+\alpha_k\, d_j)-f(x_k+\alpha_k\, d_i)-f(x_k+\alpha_k\, d_j)+f(x_k)}{\alpha_k^2}. \end{aligned} \tag{6.19}$$

(f) Compute a unitary eigenvector v_k associated with $\lambda_{\min}(H_k)$. If v_k or -v_k satisfies (6.18), declare the iteration successful with d_k equal to v_k or -v_k, otherwise declare the iteration unsuccessful.

2 - Update Step

If the iteration is successful, set $x_{k+1} = x_k + \alpha_k\, d_k$ and $\alpha_{k+1} = \min\{\gamma\, \alpha_k, \alpha_{\max}\}$; otherwise set $x_{k+1} = x_k$ and $\alpha_{k+1} = \theta\, \alpha_k$.

end

the properties of the approximate eigenvector v_k and the theoretical guarantees of every unsuccessful iteration.

Proposition 6.1 *Let v_k be the unitary eigenvector used in Algorithm 6.1. Suppose that f satisfies Assumption 6.1 and that $\lambda_{\min}(\nabla^2 f(x_k)) < 0$. Then, one has:*

$$v_k^\top \nabla^2 f(x_k)\, v_k \;\leq\; \sigma_{\min}(B_k)^2\, \lambda_k + \frac{10\, n\, \nu_H}{3} \alpha_k. \tag{6.20}$$

where $\lambda_k = \lambda_{\min}(\nabla^2 f(x_k))$.

Proof. The formulas defining the approximated Hessian H_k together with f satisfying Assumption 6.1 lead to the following error bound:

$$\forall\, (i,j) \in \{1,\ldots,n\}^2,\; \left|(H_k)_{ij} - d_i^\top \nabla^2 f(x_k)\, d_j\right| \;\leq\; \frac{5\,\nu_H}{3} \alpha_k, \tag{6.21}$$

hence

$$\left\|H_k - B_k^\top \nabla^2 f(x_k)\, B_k\right\| \;\leq\; \left\|H_k - B_k^\top \nabla^2 f(x_k)\, B_k\right\|_F \;\leq\; n\, \frac{5\,\nu_H}{3} \alpha_k, \tag{6.22}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\|\cdot\|_F$ the Frobenius norm. From this bound on the approximation error, one obtains a bound regarding the minimum eigenvalue approximation (see [38, Proposition 10.14]):

$$\left|\lambda_{\min}(H_k) - \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right)\right| \;\leq\; \frac{5\, n\, \nu_H}{3} \alpha_k. \tag{6.23}$$

Putting all together, one obtains

$$\begin{aligned}
v_k^\top \nabla^2 f(x_k)\, v_k &= \lambda_{\min}(H_k) + v_k^\top \left[\nabla^2 f(x_k) - H_k\right] v_k \\
&\leq \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) + \left|\lambda_{\min}(H_k) - \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right)\right| + \\
&\quad \|v_k\|^2\, \frac{5\, n\, \nu_H}{3}\, \alpha_k \\
&\leq \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) + (1 + \|v_k\|^2)\, \frac{5\, n\, \nu_H}{3}\, \alpha_k \\
&= \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) + \frac{10\, n\, \nu_H}{3}\, \alpha_k.
\end{aligned}$$

Since B_k is a basis of $\mathbb{R}^n$, $B_k^\top B_k$ is positive definite. For every vector $y \in \mathbb{R}^n \setminus \{0\}$, we have

$$\begin{aligned}
\frac{y^\top B_k^\top \nabla^2 f(x_k)\, B_k\, y}{\|y\|^2} &\geq \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right), \\
\frac{y^\top B_k^\top \nabla^2 f(x_k)\, B_k\, y}{\|y\|^2} \times \frac{\|y\|^2}{y^\top B_k^\top B_k\, y} &\geq \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) \frac{\|y\|^2}{y^\top B_k^\top B_k\, y}, \\
\frac{y^\top B_k^\top \nabla^2 f(x_k)\, B_k\, y}{y^\top B_k^\top B_k\, y} &\geq \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) \frac{\|y\|^2}{y^\top B_k^\top B_k\, y}.
\end{aligned}$$

Taking the minimum over all non-zero vectors in $\mathbb{R}^n$, one obtains:

$$\lambda_{\min}(\nabla^2 f(x_k)) \;\geq\; \lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) \max_{y\neq 0} \frac{\|y\|^2}{y^\top B_k^\top B_k\, y}, \tag{6.24}$$

again using the fact that B_k is a basis. Indeed, this ensures that both minimum eigenvalues have the same sign: consequently, $\lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right) < 0$ and the minimum becomes a maximum. One finally has

$$\lambda_{\min}(\nabla^2 f(x_k)) \;\geq\; \frac{\lambda_{\min}\left(B_k^\top \nabla^2 f(x_k)\, B_k\right)}{\sigma_{\min}(B_k)^2}, \tag{6.25}$$

hence the result. □

Note that in Algorithm 6.1, we do not allow for a computation of the approximate Hessian along several iterations, yet in such cases, one can still derive errors bounds that turn out to be worse than those presented above. Indeed, in a scenario where an approximation of the Hessian is computed separately along l successive iterations, one can prove that if such iterations are unsuccessful, then the error bound (6.22) becomes of order $\mathcal{O}(\theta^{-l}\, n\, \nu_H\, \alpha_k)$. This holds for a naive implementation of the method, thus these bounds are likely to be improved by considering efficient practical strategies.

Proposition 6.1 shows that the approximation error between $v_k^\top \nabla^2 f(x_k)\, v_k$ and λ_k involves the minimum singular value of a certain matrix, as well as an error of order $\mathcal{O}(n\, \nu_H\, \alpha_k)$. These elements are consistent with those obtained when using fully quadratic models (see [38, Part I] and the references therein). In fact, the square of the singular value $\sigma_{\min}(B_k)$ plays a role that is similar to the poisedness constant, hence the following assumption on those singular values.

Assumption 6.5 *The polling sets satisfy Assumption 6.4. In addition, the bases B_k are chosen such that there exists $\sigma > 0$, independent of k, such that*

$$\forall\, k, \quad \sigma_{\min}(B_k)^2 \geq \sigma. \tag{6.26}$$

When the B_k are orthonormal bases, one can choose $\sigma = \sigma_{\min}(B_k) = 1$. This is the case, for instance, when all polling sets are equal to $[Q \;\; -Q]$, with Q being an orthogonal matrix.

Lemma 6.3 *Let k be the index of an unsuccessful iteration of Algorithm 6.1 such that $\mathrm{cm}(D_k) \geq \kappa_g > 0$ and $\sigma_{\min}(B_k)^2 \geq \sigma$. Suppose that f satisfies Assumption 6.1. In that case,*

$$\|\nabla f(x_k)\| \;\leq\; \kappa_g^{-1}\left(\frac{\rho(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k\right) \tag{6.27}$$

is satisfied, and, if $\lambda_k < 0$,

$$\lambda_k \;\geq\; -\sigma^{-1}\left(\frac{2\,\rho(\alpha_k)}{\alpha_k^2} + (10\,n+1)\frac{\nu_H}{3}\alpha_k\right) \tag{6.28}$$

holds.

Proof. Equation (6.27) is obtained as in the proof of Lemma 6.2, considering that we use a PSS D_k at each iteration.

To arrive at (6.28), notice that we evaluate f at both $x_k + \alpha_k v_k$ and $x_k - \alpha_k v_k$. Thus, we can obtain the analogous of (6.4) for v_k, which is

$$-2\,\rho(\alpha_k) \;\leq\; \alpha_k^2\, v_k^\top\, \nabla^2 f(x_k)\, v_k + \frac{\nu_H}{3}\alpha_k^3. \tag{6.29}$$

Since we are in the assumptions of Proposition 6.1, we can replace the Rayleigh quotient by an expression only depending on λ_k and α_k, and we arrive at (6.28). □

We point out that for unsuccessful iterations, the corresponding Rayleigh measure is an approximation of the minimum eigenvalue with an error in $\mathcal{O}(\alpha_k)$: this is the key property that turns the weak second-order results into strong second-order ones. Thanks to this result, we obtain the following convergence theorem, whose proof follows the one of Theorem 6.1, only with λ_k playing the role of $\mathrm{rm}\,(V_k, \nabla^2 f(x_k))$.

Theorem 6.2 *We consider Algorithm 6.1 under Assumptions 2.2, 6.1, 6.2 and 6.5. Then,*

$$\liminf_{k\to\infty} \max\{\|\nabla f(x_k)\|, -\lambda_k\} \;=\; 0, \tag{6.30}$$

i.e., the method is second-order globally convergent.

This result confirms that whenever directions determined by a Hessian approximation are used, the accuracy of the approximation is the key for controlling the second-order criterion (namely the minimum Hessian eigenvalue). This has an undeniable cost in terms of function evaluations. However, if some special structure is known about the problem, then this amount of evaluations can be reduced. Random sampling can also reduce the cost if the Hessian is sparse, even if the sparsity pattern is unknown [13].

6.3.3 Complexity properties

This section is dedicated to the complexity of direct search in determining second-order stationary points. We mainly develop our reasoning for the "strong" second-order globally convergent approach of Section 6.3.2. This being said, the upcoming analysis is applicable to any second-order criterion of interest.

Our objective is to provide a bound on the number of iterations needed to ensure:

$$\inf_{0\leq l\leq k} \|\nabla f(x_l)\| < \epsilon_g \text{ and } \sup_{0\leq l\leq k} \lambda_l > -\epsilon_H, \tag{6.31}$$

given two thresholds $\epsilon_g, \epsilon_H \in (0,1)$. When (6.31) is satisfied, we say that we reached approximate second-order optimality, with respect to ϵ_g and ϵ_H. Our analysis will address first and second-order optimality simultaneously, for clarity purposes. The proof conveniently follows the argumentation of Vicente [114] for the first-order case, which we did not detail in Chapter 2.

Although the first-order result established by Vicente [114, Corollary 3.1] could still be applied to the settings of Sections 6.2 and 6.3, we treat first and second-order optimality simultaneously, for both self-containedness and clarity. The analysis establishes separate bounds on the number of successful and unsuccessful iterations needed to achieve (6.31).

For the rest of this section, we will consider a typical family of forcing functions, namely $\rho(\alpha) = \frac{c}{6}\,\alpha^q$, where $c > 0$ and $q > 2$ (those functions clearly satisfy Assumption 6.2). We start by bounding the number of successful iterations.

Theorem 6.3 *Suppose that the assumptions of Theorem 6.2 hold. Assume that Algorithm 6.1 is applied with $\rho(\alpha) = \frac{c}{6}\,\alpha^q$ with $c > 0$ and $q > 2$.*

Given $\epsilon_g, \epsilon_H \in (0,1)$, let k_0 be the index of the first unsuccessful iteration and assume that (6.31) does not hold, and let l_1 be the first index such that (6.31) is satisfied at iteration $l_1 + 1$. Then the number of successful iterations between k_0 and l_1, denoted by $|S_{l_1}(k_0)|$, is bounded as follows:

$$|S_{l_1}(k_0)| \;\leq\; \left\lceil \left(\frac{6(f(x_{l_0}) - f_{\rm low})}{c\,(\theta\,L_s)^q} \right) \max\left(\kappa_g^{-q}\,\epsilon_g^{-q}, (\sigma/n)^{-\frac{q}{\min(q-2,1)}}\,\epsilon_H^{-\frac{q}{\min(q-2,1)}} \right) \right\rceil, \quad (6.32)$$

where

$$L_s = \min\left(1, L_1^{-1}, L_2^{-\frac{1}{\min(q-2,1)}}\right),\; L_1 = \frac{c + 3\,\nu_g}{6},\; \textit{and}\; L_2 = \frac{c + 11\,\nu_H}{3}.$$

Proof. For every $l \in U$ such that $k_0 \leq l < l_1$, we know that either

$$\|\nabla f(x_l)\| \;\geq\; \epsilon_g \quad (6.33)$$

or

$$\lambda_l \;\leq\; -\epsilon_H, \quad (6.34)$$

otherwise (6.31) would have held from iteration l.

In the first case, using (6.27), we have that

$$\epsilon_g \;\leq\; \|\nabla f(x_l)\| \;\leq\; \kappa_g^{-1}\left[\frac{c}{6}\,\alpha_l^{q-1} + \frac{\nu_g}{2}\alpha_l\right].$$

Thus, if $\alpha_l < 1$,

$$\epsilon_g \;\leq\; \frac{c + 3\,\nu_g}{6\,\kappa_g}\,\alpha_l^{\min(q-1,1)}$$

and if not $\alpha_l \geq 1 > \epsilon_g$, from which we deduce

$$\alpha_l \;\geq\; \min(1, L_1^{-\frac{1}{\min(q-1,1)}})\,\kappa_g^{\frac{1}{\min(q-1,1)}}\,\epsilon_g^{\frac{1}{\min(q-1,1)}}.$$

Since $q > 2$, this reduces to

$$\alpha_l \;\geq\; \min(1, L_1^{-1})\,\kappa_g\,\epsilon_g. \quad (6.35)$$

In the second case, we obtain from (6.28) that

$$-\epsilon_H \;\geq\; \lambda_l \;\geq\; -\sigma^{-1}\left(\frac{c}{3}\,\alpha_l^{q-2} + \frac{(10\,n+1)\,\nu_H}{3}\,\alpha_l\right) \;\geq\; -\sigma^{-1}\,n\,\left(\frac{c}{3}\,\alpha_l^{q-2} + \frac{11\,\nu_H}{3}\,\alpha_l\right),$$

which leads by the same reasoning as above to

$$\alpha_l \;\geq\; \min(1, L_2^{-\frac{1}{\min(q-2,1)}})\,(\sigma/n)^{\frac{1}{\min(q-2,1)}}\,\epsilon_H^{\frac{1}{\min(q-2,1)}}. \tag{6.36}$$

As a result of (6.35) and (6.36), for all unsuccessful iterations of index $k_0 \leq l < l_1$, one has the following lower bound on the step size

$$\alpha_l \;\geq\; L_s\,\min\left(\kappa_g\,\epsilon_g, (\sigma/n)^{\frac{1}{\min(q-2,1)}}\,\epsilon_H^{\frac{1}{\min(q-2,1)}}\right).$$

Consider now the successful iterations of index k, $k_0 < k \leq l_1$. For each iteration k of this type, one can backtrack to the previous unsuccessful iteration (which exists since $k_0 \in U$), denoted by $l(k)$, such that $\alpha_k \geq \theta\,\alpha_{l(k)}$, given the update rules for the step size parameter. Thus, for any of those iterations, one has:

$$\alpha_k \;\geq\; \theta\,L_s\,\min\left(\kappa_g\,\epsilon_g, (\sigma/n)^{\frac{1}{\min(q-2,1)}}\,\epsilon_H^{\frac{1}{\min(q-2,1)}}\right), \tag{6.37}$$

and by definition of a successful iteration:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \rho(\alpha_k) \\ &\geq \frac{c}{6}\,(\theta\,L_s)^q\,\min\left(\kappa_g^q\,\epsilon_g^q, (\sigma/n)^{\frac{q}{\min(q-2,1)}}\,\epsilon_H^{\frac{q}{\min(q-2,1)}}\right). \end{aligned}$$

Thus, by summing on all successful iterations until l_1 excluded, we arrive at

$$f(x_{k_0}) - f(x_{l_1}) \;\geq |S_{l_1}(k_0)|\,\frac{c}{6}\,(\theta\,L_s)^q\,\min\left(\kappa_g^q\,\epsilon_g^q, (\sigma/n)^{\frac{q}{\min(q-2,1)}}\,\epsilon_H^{\frac{q}{\min(q-2,1)}}\right).$$

and the result stated in the theorem follows from Assumption 2.2. □

We then treat the case of the unsuccessful iterations, using the simplified analysis of Garmanjani, Júdice and Vicente [60].

Theorem 6.4 *Let the assumptions of Theorem 6.3 hold. Then, with the same definitions for k_0 and l_1, the number of unsuccessful iterations between k_0 and l_1 is at most $|U_{l_1}(k_0)|$, where*

$$|U_{l_1}(k_0)| \;\leq\; \left\lceil L_3\,|S_{l_1}(k_0)| + L_4 - \log_\theta e\,\max\left\{\kappa_g^{-1}\,\epsilon_g^{-1}, (\sigma/n)^{-\frac{1}{\min(q-2,1)}}\,\epsilon_H^{-\frac{1}{\min(q-2,1)}}\right\}\right\rceil \tag{6.38}$$

with

$$L_3 = -\log_\theta \gamma \quad \textit{and} \quad L_4 = \log_\theta\left(\frac{\theta\,L_s\,e}{\alpha_{k_0}}\right).$$

Proof. By induction, one has:

$$\alpha_{l_1} \;\leq\; \alpha_{k_0}\,\gamma^{|S_{l_1}(k_0)|}\,\theta^{|U_{l_1}(k_0)|},$$

which, as $\theta \in (0,1)$, leads to

$$|U_{l_1}(k_0)| \;\leq\; -\log_\theta \gamma\,|S_{l_1}(k_0)| - \log_\theta \alpha_{k_0} + \log_\theta \alpha_{l_1}. \tag{6.39}$$

Since $\ln\theta < 0, \ln\gamma > 0$, and α_{l_1} is bounded below from (6.37), (6.39) becomes

$$|U_{l_1}(k_0)| \;\leq\; L_3\,|S_{l_1}(k_0)| + \log_\theta\left(\frac{\theta\,L_s}{\alpha_{k_0}}\right) - \frac{\ln\left(\max\left\{\kappa_g^{-1}\,\epsilon_g^{-1}, (\sigma/n)^{-\frac{1}{\min(q-2,1)}}\,\epsilon_H^{-\frac{1}{\min(q-2,1)}}\right\}\right)}{\ln\theta}.$$

Finally, we apply $\ln(x) \leq x - 1$ and arrive at the desired result. □

As explained in [114], the index of the first unsuccessful iteration k_0 can be bounded from above, thanks to Assumption 2.2. In our case, we can choose the following quantity as an upper bound:

$$\left\lceil \frac{6(f(x_0) - f_{\text{low}})}{c\,\alpha_0^q}\,\max\left(\kappa_g^{-q}\,\epsilon_g^{-q}, (\sigma/n)^{-\frac{q}{\min(q-2,1)}}\,\epsilon_H^{-\frac{q}{\min(q-2,1)}}\right)\right\rceil.$$

This leads to the following result regarding approximate second-order optimality.

Theorem 6.5 *Let the assumptions of Theorem 6.3 hold. The number of iterations needed by Algorithm 6.1 to satisfy* (6.31) *is at most*

$$\mathcal{O}\left(\max\left(\kappa_g^{-q}\,\epsilon_g^{-q}, (\sigma/n)^{-\frac{q}{\min(q-2,1)}}\,\epsilon_H^{-\frac{q}{\min(q-2,1)}}\right)\right), \tag{6.40}$$

where the constant in $\mathcal{O}(\cdot)$ depends on $\nu_g, \nu_H, \alpha_0, f(x_0), f_{\text{low}}, c, \gamma, \theta$, and q.

The best power of ϵ_H (that is, the least negative power) is here achievable choosing $q = 3$.

We now give the corresponding result with respect to the number of function evaluations.

Theorem 6.6 *Under the assumptions of Theorem 6.3, the number of function evaluations needed by Algorithm 6.1 is at most*

$$\mathcal{O}\left(m\,\max\left(\kappa_g^{-q}\,\epsilon_g^{-q}, (\sigma/n)^{-\frac{q}{\min(q-2,1)}}\,\epsilon_H^{-\frac{q}{\min(q-2,1)}}\right)\right), \tag{6.41}$$

with m is the maximum number of function evaluations performed in any iteration. Here again, the constant in $\mathcal{O}(\cdot)$ only depends on $\nu_g, \nu_H, \alpha_0, f(x_0), f_{\text{low}}, c, \gamma, \theta$, and q.

As in the first-order case [26, 48, 60, 114], we are interested in the order of n that appears in the complexity bounds related to the number of function evaluations. We will see that such an order is considerably higher than in the first-order case, which is not surprising given the requirements we impose on the polling sets.

The value of m in (6.41) depends on the choice of the polling sets, their cardinality and whether they have a non empty symmetric part. For instance, $D_k = D_\oplus = [I \ \ -I]$ leads to at most

$$m = 2\,n + \frac{n^2 - n}{2} + 2 = \frac{n^2 + 3\,n + 4}{2}$$

evaluations by using orthonormal bases included in $D_\oplus$. Given that $\mathrm{cm}(D_\oplus) = 1/\sqrt{n}$, one can then replace κ_g by $1/\sqrt{n}$ and σ by 1. When $q = 3$, κ_g^{-q} becomes $n^{\frac{3}{2}}$, thus less than n^3, showing that the second-order part dominates the power of n in (6.40). The dependence of (6.41) on n, when using $D_\oplus$, is of the order n^5.

Corollary 6.2 *Consider the application of Algorithm 6.1, under the assumptions of Theorem 6.3 with $q = 3$. Suppose D_k is chosen as $D_\oplus$ for all k (or as any other PSS D such that $m = \mathcal{O}(n^2)$, $\mathrm{cm}(D) = \mathcal{O}\,(1/\sqrt{n})$, and the B_k's are orthogonal matrices). Then, to satisfy* (6.31)*, the method takes at most*

$$\mathcal{O}\left(\max\left\{n^{\frac{3}{2}}\,\epsilon_g^{-3}, n^3\,\epsilon_H^{-3}\right\}\right) \tag{6.42}$$

iterations and

$$\mathcal{O}\left(\max\left\{n^{\frac{7}{2}}\,\epsilon_g^{-3}, n^5\,\epsilon_H^{-3}\right\}\right) \tag{6.43}$$

function evaluations, where the constant in $\mathcal{O}(\cdot)$ only depends on $\nu_g, \nu_H, \alpha_0, f(x_0), f_{\mathrm{low}}, c$, γ, and θ.

As explained in Section 6.3.2, preliminary knowledge regarding the structure of the Hessian may help reducing the powers of n.

Our analysis covers a wide class of direct-search algorithms. Note that it can be simplified following the process of Konečný and Richtárik [78] in the case where the step size is never increased and it is halved at every unsuccessful iteration (i.e., $\gamma = 1$ and $\theta = 1/2$). Choosing $\rho(\alpha) = \alpha^3$ (hence $q = 3$) as well as $D_k = D_\oplus$ for all iterations (as they provide the best known bounds), one could easily see that the number of successful iterations becomes:

$$|S_{l_1}(k_0)| \ \leq \ \left\lceil \left(\frac{f(x_{l_0}) - f_{\mathrm{low}}}{8\,L_s^3}\right) \max\left(n^{\frac{3}{2}}\,\epsilon_g^{-3}, n^3\,\epsilon_H^{-3}\right)\right\rceil, \tag{6.44}$$

by the same reasoning as in the proof of Theorem 6.3. On the other hand, since $\gamma = 1$ it would be much simpler to bound the number of unsuccessful iterations. The result corresponding to (6.38) is

$$|U_{l_1}(k_0)| \ \leq \ \left\lceil \log_2(\alpha_{k_0}) - \log_2\left(\frac{L_s}{2}\min\left\{\frac{\epsilon_g}{\sqrt{n}}, \frac{\epsilon_H}{n}\right\}\right)\right\rceil. \tag{6.45}$$

The conclusions of Corollary 6.2 are then unchanged in terms of dependences on n,ϵ_g, and ϵ_H.

To the best of our knowledge, the above results were the first complexity bounds to have been established regarding the determination of second-order stationary points by a derivative-free method. It is interesting to compare them to the existing second-order complexity bounds previously obtained in the derivative-based literature. Cartis et al [25] derived such bounds for adaptive regularization with cubics (ARC) and trust-region methods, respectively in

$$\mathcal{O}\left(\max\left\{\epsilon_g^{-\frac{3}{2}}, \epsilon_H^{-3}\right\}\right) \quad \text{and} \quad \mathcal{O}\left(\max\left\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\right\}\right).$$

They proved also that whenever $\epsilon_H \leq \epsilon_g$, the two bounds reduce to $\mathcal{O}(\epsilon_H^{-3})$, and gave an example to show that such a bound is sharp in terms of this tolerance. Our bound also reduces to $\mathcal{O}(\epsilon_H^{-3})$ in that case, so we may say that the three bounds are comparable from such a point of view; in our case, the sharpness of the bound remains to be proved.

When keeping both tolerances ϵ_g and ϵ_H in the bounds, we see that the first part of our bound is worse than the one obtained by ARC, whereas it seems comparable to the one for trust-region methods. However, we obtain ϵ_g^{-3} instead of $\epsilon_g^{-2}\,\epsilon_H^{-1}$. This discrepancy is related to the decrease requirements. In trust-region methods, provided (6.31) is not satisfied, one has

$$f(x_k) - f(x_{k+1}) \;\geq\; \eta \,\min\left\{\mu_1\,\epsilon_g\,\delta_k, \mu_2\,\epsilon_H\,\delta_k^2\right\} \quad \text{and} \quad \delta_k \geq \mu_3\,\min\{\epsilon_g, \epsilon_H\}, \tag{6.46}$$

where $\eta, \mu_1, \mu_2, \mu_3$ are positive constants and δ_k is the trust-region radius, which is the key argument to prove the complexity results [25, Section 4]. Two decreases are considered in (6.46), concerning respectively first and second-order criteria. In our direct-search frameworks, we only have one decrease formula that depends on α_k^3, hence the discrepancy.

In the derivative-free community, we point out a recent PhD thesis [73] in which the author proposes a complexity bound for second-order optimality. These results match ours in terms of power of the tolerances. However, we were not able to find second-order complexity results for derivative-free equivalents of ARC methods [26] in the existing literature. Still, it is our belief that these results would match those in [25] regarding the powers of the tolerances, again because of the decrease formulas that are employed to prove the convergence.

6.4 Numerical study of the second-order framework

The techniques we describe guarantee that the algorithms are able to exploit negative curvature whenever needed. For instance, we can apply Algorithm 6.1 to the function f_1 described in Section 6.2.2, starting at the origin with the polling set equal to $D_{\oplus}$. As we have seen, none of the polling directions will yield a decrease in the objective function, because the corresponding values in those directions are always positive. However, if

we compute an approximate Hessian and its eigenvector associated with the minimum eigenvalue, we obtain a direction in which the function has a negative value. With a sufficiently small step size, it will satisfy the sufficient decrease condition.

Besides ensuring to escape saddle points and local maximizers, the use of negative curvature is also known to improve the practical performance of line-search methods [63, 86, 102], as going in negative curvature directions possibly leads to faster decrease in the objective function value. We are interested in knowing if our approach allows for the same kind of improvement. To this end, we compare our implementation of Algorithm 6.1 (`ahds`) to a basic implementation of Algorithm 2.1 (thereafter called `bds`, for basic direct search).

We consider less pathological problems than the one mentioned above. Those functions were taken from the `CUTEst` package [65], and have been identified as presenting negative curvature at some points by Avelino et al. [12, Table 6]. This represents a total of 60 problems out of the 119 problems tested in [12]. For all of those problems, we used the smallest dimension available in the SIF files, resulting in 36 problems with dimensions less than 10, 22 problems having dimensions between 10 and 15, 1 problem with dimension 28, and 1 problem with dimension 50. The full list of problems is given in Appendix B.

We tested four types of polling sets choices for Step 1 in Algorithm 6.1. The two first choices correspond to the set $D_{\oplus}$ and the minimal positive basis with uniform angles we defined in Example 2.1, denoted by V_{n+1}. These are common choices in practice, and also typical examples of polling sets, with and without symmetric parts, respectively. The two other types of PSSs are based on $D_{\oplus}$ and V_{n+1}, built by applying a rotation matrix Q to those sets. All of our choices keep D_k constant throughout all iterations. However, to measure the effect of our approach on a larger variety of sets, we ran the method ten times, varying the ordering of the vectors for the choices $\{D_{\oplus}, V_{n+1}\}$, or changing the rotation matrix Q in the other cases. Table 6.1 summarizes our polling set choices for clarity; in the rest of the section, we will identify a method as `bds`i or `ahds`i with $i \in \{0, 1, 2, 3\}$ indicating the polling choice.

Polling Number	Set type	Cardinality	Variant
0	$D_{\oplus}$	$2\,n$	Ordering of directions
1	$Q\,D_{\oplus}$	$2\,n$	Rotation matrix Q
2	V_{n+1}	$n+1$	Ordering of directions
3	$Q\,V_{n+1}$	$n+1$	Rotation matrix Q

Table 6.1: The different polling set choices for Algorithm 6.1.

For all methods, the forcing function was $\rho(\alpha) = 10^{-3}\,\alpha^3$, the starting point was the one defined by `CUTEst`, and the starting step size was $\alpha_0 = 1$. We consider that a run is successful whenever the final function value f_* satisfies:

$$f_* - f_{best} < \epsilon\,(f(x_0) - f_{best})\,, \tag{6.47}$$

where f_{best} is the best value obtained by all variants with an extended budget of $5000\,n$ iterations, and $\epsilon > 0$ is a given tolerance [95]. For each method, it is plotted a performance profile [50] for the average number of function evaluations taken on the 10 runs.

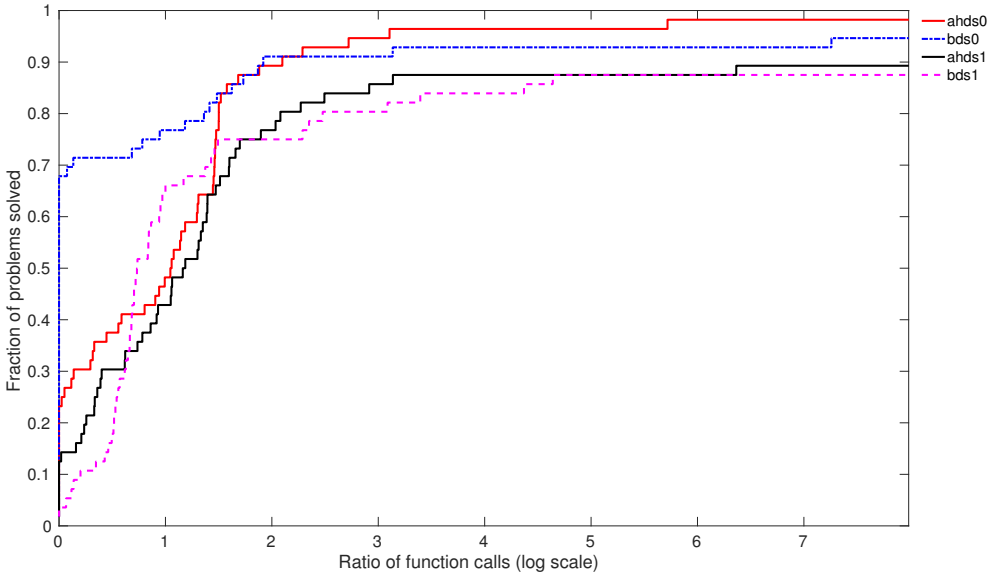
The results we present illustrate the typical behavior of Algorithm 6.1 compared to the classic direct-search scheme of Algorithm 2.1. Figure 6.1 firstly shows profiles obtained for the symmetric polling choices 0/1. One sees that the methods `bds0` and `bds1` perform still better than `ahds0` and `ahds1` in terms of efficiency (ratio=0). However, the `ahds` methods eventually solve more problems than the `bds` ones (large ratio), thus being more robust. This tendency was to be expected, as second-order mechanisms help in exploiting negative curvature.

Figure 6.2 is related to the non-symmetric polling set choices, and is characteristic of the performances of both methods. For such polling strategies, the gain from the `bds` to the `ahds` methods is even higher than in the symmetric case and now also present in efficiency. However, when one looks at the directions that lead to successful iterations for the `ahds` methods, one sees that half of the successful iterations are successful because of a direction within the original PSS D_k (step (a) of the polling), one third are successful thanks to a direction in $-D_k$ (step (b)), and only around 10% succeed with the approximate eigenvector v_k (step (f)). Our interpretation is that considering opposite directions already increases the chances to exploit negative curvature (as it guarantees weak second-order global convergence), while allowing to poll along additional directions in the case of a non-symmetric PSS.

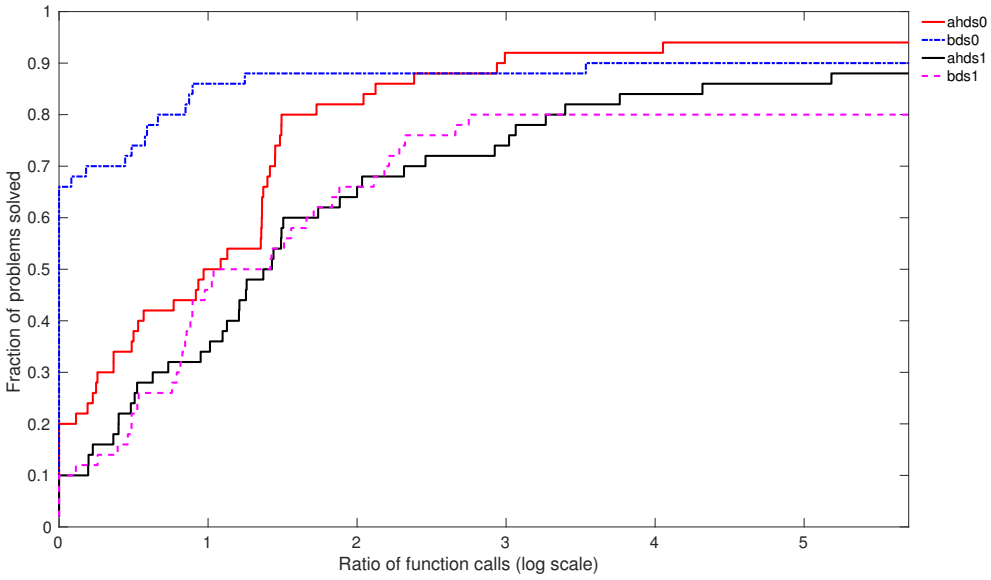
To support this hypothesis, we implemented a variant of Algorithm 6.1 that did not attempt to build an approximate Hessian (we thus lost second-order convergence to weak second-order convergence). Such a variant, called `sds`, already outperformed its basic direct search counterpart on the tested problems, yet the corresponding `ahds` algorithm stood out as the most robust implementation.

Figures 6.3 and 6.4 present the results for the polling strategies 2/3. One observes that the `sds` method is generally more efficient than the corresponding `bds` and `ahds` instances, the only exception being when $\epsilon = 10^{-6}$ and $D_k = V_{n+1}$. In this particular setting, the `ahds` method outperforms the other two, and the reason appears to be that the amount of successful iterations corresponding to the last step of the method is significantly higher than in the other settings. The conclusions of Figure 6.1 can thus be extended for the non-symmetric case: the `ahds` algorithm eventually benefits from the computation of an approximate Hessian eigenvector. These profiles promote the use of symmetric positive spanning sets as a first attempt to catch negative curvature information, and confirm that curvature has even more chance of being exploited by computing a Hessian approximation.

A final comment can be made by studying the relative performance among different instances of Algorithm 6.1 (see Figure 6.5). The method `ahds0` solves the most problems within the given budget, and also outperforms the other variants. Using symmetric positive spanning sets and building Hessian approximation with respect to orthonormal bases thus seems to allow exploiting more second-order information in general. Note

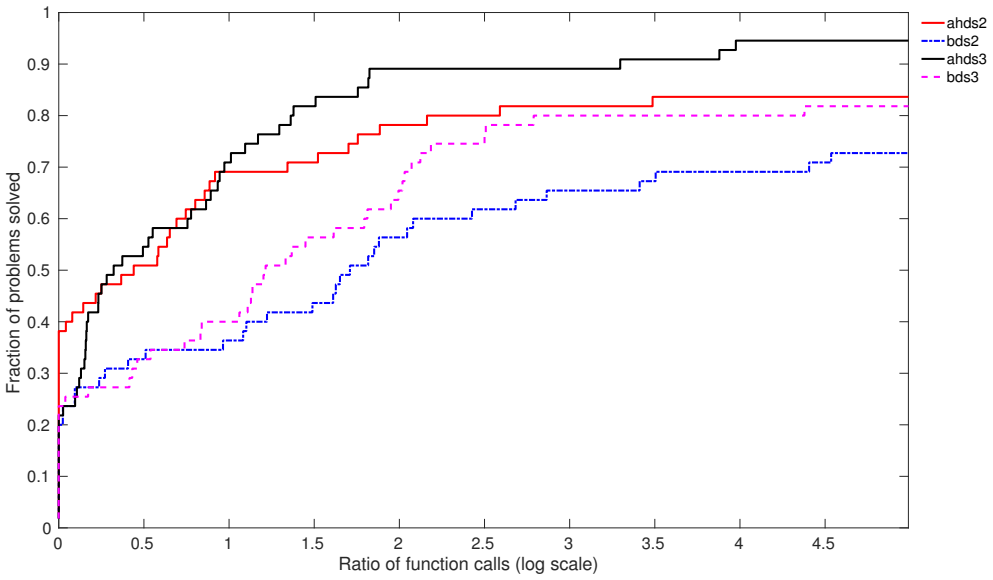


(a) $\epsilon = 10^{-3}$.

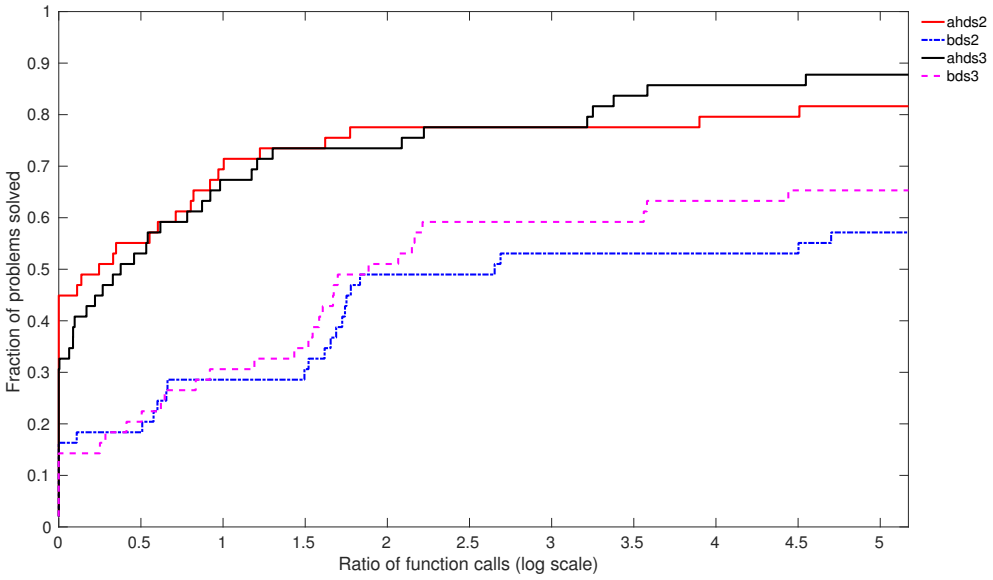


(b) $\epsilon = 10^{-6}$.

Figure 6.1: Performance of the methods with polling choices 0/1, given a budget of $2000\,n$ evaluations.

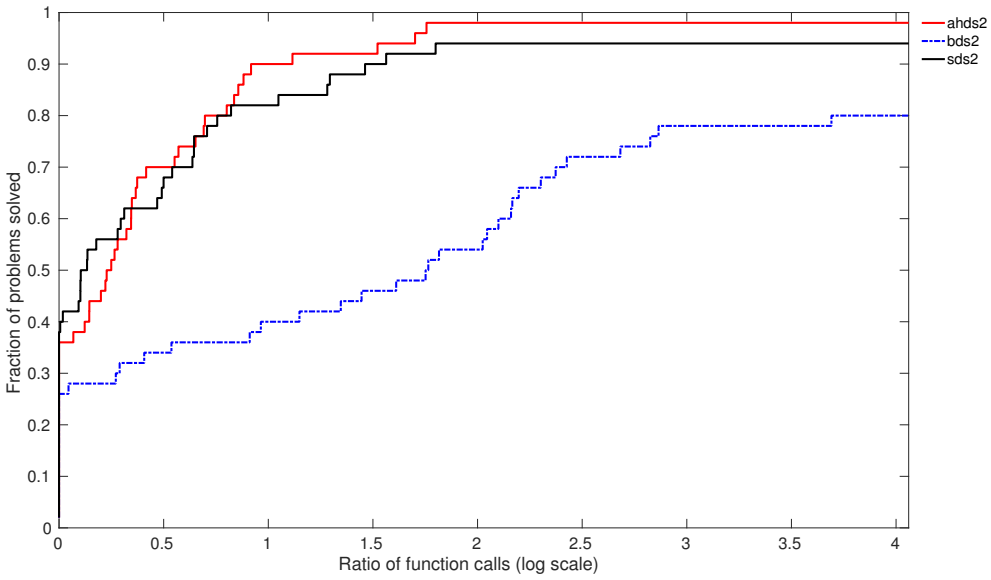


(a) $\epsilon = 10^{-3}$.

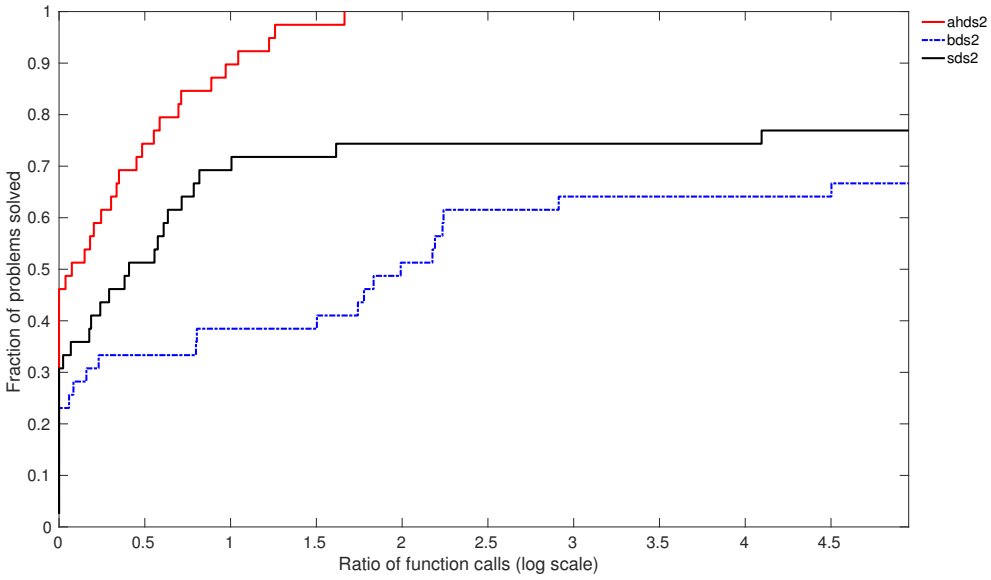


(b) $\epsilon = 10^{-6}$.

Figure 6.2: Performance of the methods with polling choices 2/3, given a budget of $2000\,n$ evaluations.

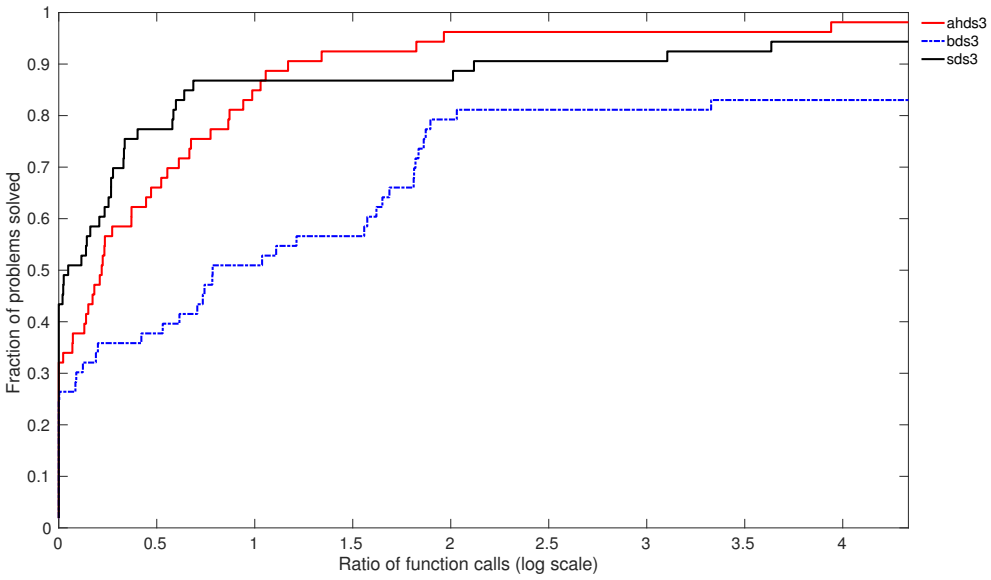


(a) $\epsilon = 10^{-3}$.

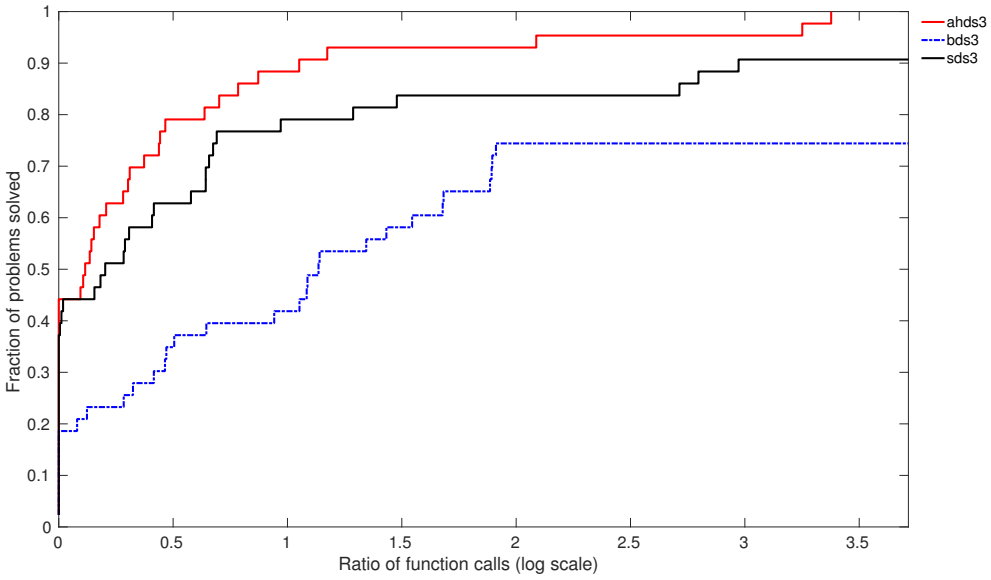


(b) $\epsilon = 10^{-6}$.

Figure 6.3: Second-, first- and weakly second-order direct-search methods, with polling choice 2 and a budget of $2000\,n$ evaluations.

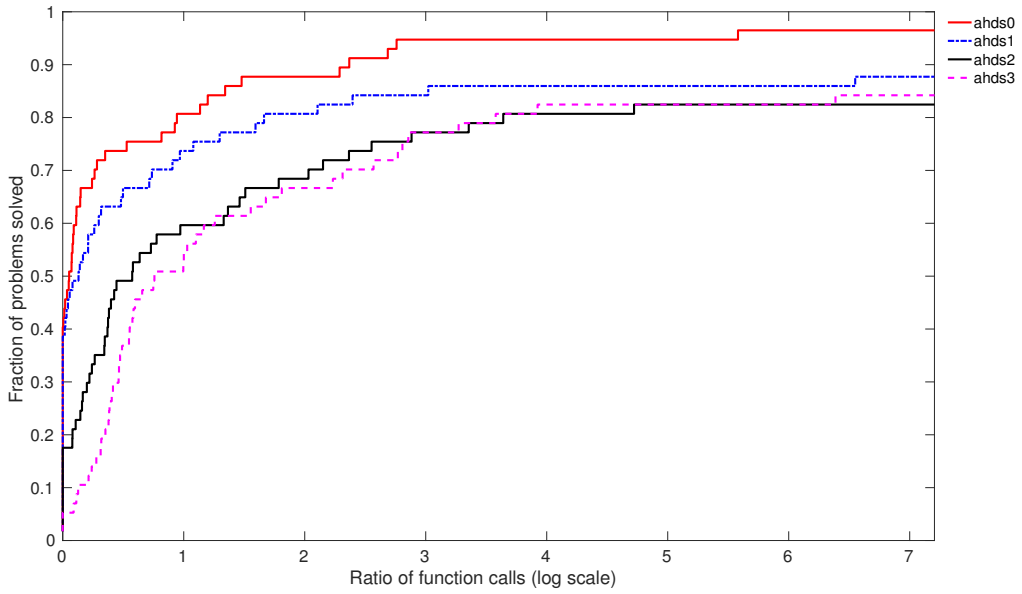


(a) $\epsilon = 10^{-3}$.

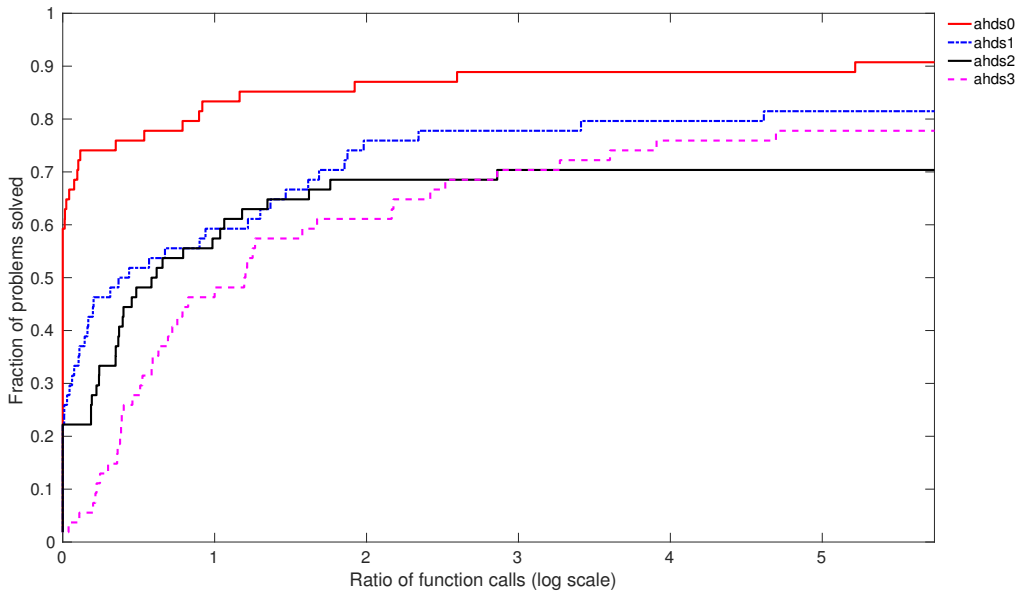


(b) $\epsilon = 10^{-6}$.

Figure 6.4: Second-, first- and weakly second-order direct-search methods, with polling choice 3 and a budget of $2000\,n$ evaluations.



(a) $\epsilon = 10^{-3}$.



(b) $\epsilon = 10^{-6}$.

Figure 6.5: Comparison of the `ahds` methods, given a budget of $2000\,n$ evaluations.

again that the weak second-order properties of the polling set are particularly relevant, and likely to provide curvature information. As we have seen, completing the polling with a phase of Hessian approximation is an additional tool that appears to pay off in the long run.

6.5 Conclusion of Chapter 6

As shown in this chapter, weak second-order optimality can be established for a broad class of direct-search methods. Indeed, one can identify a measure of the second-order aspects that are captured by the algorithm, which tends to be nonnegative. This already enables to classify the polling choices by their second-order relevance: it then appears that symmetric sets may already help on problems where curvature is worth considering. In order to design second-order globally convergent methods, one then needs to extend this measure to turn it into a true second-order one. This was done by extending the polling process to build a Hessian approximation, thus resulting in a provably second-order convergent direct-search instance. The corresponding complexity analysis reveals a dependence on the problem dimension worsened compared to the first-order case, yet this expense is comparable to the cost of other second-order globally convergent derivative-free schemes.

Despite pathological examples showing that the expense per iteration can be quite important for those algorithms, it seems that the actual practical cost remains at a reasonable level, albeit potentially higher than for typical first-order methods. The interest in this increased consumption lies in the benefit of exploiting negative curvature in practice, which can allow to obtain better solutions while increasing the robustness of the methods.

As a conclusion, we point out that the material presented in this chapter has previously appeared in *Optimization* [66]. In particular, its first appearance in the associated technical report and preprint was anterior to the results in [73], which justifies our claim that to the best of our knowledge, the second-order complexity bounds derived in this chapter were the first of their kind to be established in the derivative-free literature.

Chapter 7

De-coupled first/second-order steps strategies for nonconvex derivative-free optimization

In order to guarantee second-order convergence, optimization methods must account for two criteria that might not be of the same order of magnitude, while this difference in scaling might vary throughout the algorithmic process. Whenever not carefully handled, such a behavior may prevent fast convergence of the algorithm, and even cancel any benefits that can come from considering second-order aspects. Consequently, any framework with global (second-order) convergence purposes should ideally be able to identify the potential scaling differences, and act appropriately to prevent those from impacting practical efficiency..

In this chapter, we propose a technique that transforms an algorithm so as to dissociate its first and second-order features. The idea is widely applicable to any provably second-order globally convergent scheme, and gives rise to new methods we qualify as *de-coupled*. After describing further the general concept in Section 7.1, we first study a derivative-free trust-region method in which the de-coupling process lies in the use of two trust-regions: this is the subject of Section 7.2. We then propose a de-coupled direct-search method in Section 7.3, showing that it enjoys the same convergence guarantees as the algorithm introduced in Chapter 6, but with improved complexity results. Introduction of randomness into those frameworks is discussed in Section 7.4. We conclude this chapter by numerical comparison of the de-coupled variants with their traditional "coupled" counterparts in Section 7.5.

7.1 Motivations

Our analysis is mostly inspired by was has been done in derivative-based line-search methods [12, 63, 94, 102]. In those works, it has been identified that directions of descent, that relate to the properties of the gradient, and directions of negative curvature, that are connected to the eigenvalues of the Hessian matrix, should not necessarily be associated with the same steplength. Indeed, it may be that one of the two criteria of interest (namely the gradient norm and the minimum Hessian eigenvalue) is several orders of magnitude smaller than the other. In that situation, a method based on a unique step length may compute a very small step to cope with the magnitude of one criterion, even though more improvement could have been realized by performing a moderate step by concentrating on the other one. Tests based on comparing Taylor model values at promising directions [63] or scaling techniques prior to the actual step computation [12] have been studied. In trust-region methods, those issues can be handled internally by the subproblem solver [33], yet they remain of essence considering that any step in this subproblem is limited in size by the trust-region radius. It may be that this radius is forced to shrink in order to provide second-order guarantees.

This is particularly true in derivative-free optimization, as none of the derivatives is known. The step size (in direct search) or the trust-region radius (in trust region) are often the tool of essence to estimate a joint optimality criterion (in general the maximum of the gradient norm and the opposite of the minimum eigenvalue). As a result, the cost of the second-order guarantees often absorbs the first-order ones (see [37] and Chapter 6).

Our study revolves around the separate treatment of gradient-type and Hessian-type properties of the function at a given iterate. We define a *de-coupling process* as a strategy that relies on duplicating elements of the algorithm that intervene in the treatment of both aspects, so as to treat each of them separately. As we will see, the idea is general enough to be embedded in a wide range of optimization algorithms.

7.2 A trust-region method with de-coupling process

We begin by applying the de-coupling idea to derivative-free methods of model-based type. A trust-region algorithm is presented, that relies on dissociating first and second-order contributions in the decrease obtained in the function values by means of two trust-regions.

Beforehand, we recall the general form of a second-order globally convergent trust-region method (which we consider in a derivative-free setting) in Algorithm 7.1. As in Chapter 4, we consider a simplified framework close to the probabilistic case [14], while allowing for a complete new sampling at each iteration, using a typical well-poised set of directions in order to cope with the lack of criticality step. This being said, we believe that an extension of the upcoming analysis to take such a step into account is rather straightforward, given the existing literature [60].

Algorithm 7.1: Basic Second-Order Derivative-Free Trust-Region framework

Define positive parameters η_1, η_2, $0 < \gamma_1 < 1 < \gamma_2$ and $0 < \delta_0 < \delta_{\max}$.
for $k = 0, 1, 2, \ldots$ **do**
  Approximate the function f in $B(x_k, \delta_k)$ by
  $m_k(x_k + s) = f(x_k) + g_k^\top s + \frac{1}{2} s^\top H_k s$.
  Compute s_k by approximately minimizing m_k in $B(x_k, \delta_k)$, and let

$$\rho_k \;=\; \frac{f(x_k) - f(x_k + s_k)}{m(x_k) - m(x_k + s_k)}.$$

  If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$, and

$$\delta_{k+1} \;=\; \begin{cases} \min\{\gamma_2\, \delta_k, \delta_{\max}\} & \text{if } \tau_k \geq \eta_2\, \delta_k \\ \gamma_1\, \delta_k & \text{otherwise,} \end{cases}$$

  where $\tau_k = \max\{\|g_k\|, -\lambda_{\min}(H_k)\}$.
  Otherwise, set $x_{k+1} = x_k$ and $\delta_{k+1} = \gamma_1\, \delta_k$.
end

7.2.1 A de-coupled trust-regions approach

We present below a de-coupled variant of Algorithm 7.1. In the upcoming algorithm, called DESTRESS for DE-coupled Steps in a Trust-REgionS Strategy, multiple variables and parameters will appear in both contexts of first-order trust-region and second-order trust-region. To avoid confusion between those parameters and the potential subscripts they may have, we adopt the following notation: a parameter or variable of the method will be affected the superscript or subscript "c" (Cauchy step) when relative to the first-order trust-region, and "e" (eigenstep) when related to the second-order trust-region.

The algorithmic principles of Algorithm 7.2 are highly similar to that of a classical trust-region method. The main difference is that a first-order step is computed, tested, and accepted if it yields the desired improvement. *Only when this step is rejected* are a second-order model and its associated second-order step computed. In that sense, our framework prioritizes the first-order aspects, while potentially avoiding the expense of computing the second-order model.

Both models are chosen as quadratic functions (not necessarily identical), that satisfy the following assumptions.

Assumption 7.1 *Throughout an execution of Algorithm 7.2, the sequence of first-order models $\{m_k^{\mathrm{c}}\}_k$ is a $(\kappa_g^{\mathrm{c}}, \kappa_f^{\mathrm{c}})$-fully linear model sequence for the first-order trust-region sequence, i.e., each model m_k^{c} is $(\kappa_g^{\mathrm{c}}, \kappa_f^{\mathrm{c}})$-fully linear on $B(x_k, \delta_k^{\mathrm{c}})$.*

In addition, the first-order model Hessians are bounded in norm, i.e., it exists $B_{\mathrm{c}} > 0$ such that

$$\forall\, k, \quad \|H_k^{\mathrm{c}}\| \;\leq B_{\mathrm{c}}.$$

Algorithm 7.2: DE-coupled Steps in a Trust-REgionS Strategy (DESTRESS)

Choose $x_0 \in \mathbb{R}^n,\ 0 < \delta_0^{\rm c} < \delta_{\max}^{\rm c},\ 0 < \delta_0^{\rm e} < \delta_{\max}^{\rm e},\ 0 < \gamma_1^{\rm c} < 1 < \gamma_2^{\rm c},$
$0 < \gamma_1^{\rm e} < 1 < \gamma_2^{\rm e},\ (\eta_1^{\rm c}, \eta_2^{\rm c}, \eta_1^{\rm e}, \eta_2^{\rm e}) \in (0, \infty)$.
for $k = 0, 1, 2, \ldots$ **do**

1. **First-order trust region and step**

   (a) Compute a model $m_k^{\rm c}$ of the function f and compute a step $s_k^{\rm c}$ by approximately solving the first-order trust-region subproblem

   $$\begin{cases} \min_s\ m_k^{\rm c}(x_k + s) = f(x_k) + [g_k^{\rm c}]^\top s + \frac{1}{2} s^\top H_k^{\rm c}\, s \\ \|s\| \leq \delta_k^{\rm c}. \end{cases} \tag{7.1}$$

   (b) Compute $f(x_0 + s_k^{\rm c})$ and $\rho_k^{\rm c} = \frac{f(x_k + s_k^{\rm c}) - f(x_k)}{m_k^{\rm c}(x_k + s_k^{\rm c}) - m_k^{\rm c}(x_k)}$.

   (c) If $\rho_k^{\rm c} \geq \eta_1^{\rm c}$, set $x_{k+1} = x_k + s_k^{\rm c}$ and

   $$\delta_{k+1}^{\rm c} \ = \ \begin{cases} \min\{\gamma_2^{\rm c}\, \delta_k^{\rm c}, \delta_{\max}^{\rm c}\} & if\ \|g_k^{\rm c}\| \geq \eta_2^{\rm c}\, \delta_k^{\rm c}, \\ \delta_k^{\rm c} & otherwise, \end{cases}$$

   and skip the rest of the iteration. Otherwise, go to 2.

2. **Second-order trust region and step**

   (a) Compute a model $m_k^{\rm e}$ of the function f and compute a step $s_k^{\rm e}$ by approximately solving the second-order trust-region subproblem

   $$\begin{cases} \min_s\ m_k^{\rm e}(x_k + s) = f(x_k) + [g_k^{\rm e}]^\top\, s + \frac{1}{2} s^\top H_k^{\rm e}\, s \\ \|s\| \leq \delta_k^{\rm e}. \end{cases} \tag{7.2}$$

   (b) Compute $f(x_k + s_k^{\rm e})$ and $\rho_k^{\rm e} = \frac{f(x_k + s_k^{\rm e}) - f(x_k)}{m_k^{\rm e}(x_k + s_k^{\rm e}) - m_k^{\rm e}(x_k)}$.

   (c) If $\rho_k^{\rm e} \geq \eta_1^{\rm e}$, set $x_{k+1} = x_k + s_k^{\rm e}$ and

   $$\delta_{k+1}^{\rm e} \ = \ \begin{cases} \min\{\gamma_2^{\rm e}\, \delta_k^{\rm e}, \delta_{\max}^{\rm e}\} & if\ -\lambda_k^{\rm e} = -\lambda_{\min}(H_k^{\rm e}) \geq \eta_2^{\rm e}\, \delta_k^{\rm e}, \\ \delta_k^{\rm e} & otherwise, \end{cases}$$

   and skip the rest of the iteration. Otherwise, go to 3.

3. **Unsuccessful iteration**: set

   $$\begin{cases} x_{k+1} &=& x_k, \\ \delta_{k+1}^{\rm c} &=& \gamma_1^{\rm c}\, \delta_k^{\rm c}, \\ \delta_{k+1}^{\rm e} &=& \gamma_1^{\rm e}\, \delta_k^{\rm e}. \end{cases}$$

Assumption 7.2 *Throughout an execution of Algorithm 7.2, the sequence of second-order models $\{m_k^{\mathrm{e}}\}_k$ is a $(\kappa_H^{\mathrm{e}}, \kappa_g^{\mathrm{e}}, \kappa_f^{\mathrm{e}})$-fully quadratic model sequence for the second-order trust-region sequence, i.e., each model m_k^{e} is $(\kappa_H^{\mathrm{e}}, \kappa_g^{\mathrm{e}}, \kappa_f^{\mathrm{e}})$-fully quadratic on $B(x_k, \delta_k^{\mathrm{e}})$.*

One sees that the de-coupling process results in separate requirements regarding the quality of the models.

7.2.2 Convergence and complexity analysis

We begin by assuming that the model minimization satisfies the classical properties for convergence (although we emphasize that they are usually assumed on a unique step).

Assumption 7.3 *At each iteration k of Algorithm 7.2, the model m_k^{c} is minimized by computing an* approximate Cauchy point*, i.e., the first-order step satisfies:*

$$m_k^{\mathrm{c}}(x_k) - m_k^{\mathrm{c}}(x_k + s_k^{\mathrm{c}}) \;\geq\; \frac{\tau_{\mathrm{c}}}{2} \|g_k^{\mathrm{c}}\| \min\left\{ \frac{\|g_k^{\mathrm{c}}\|}{\|H_k^{\mathrm{c}}\|}, \delta_k^{\mathrm{c}} \right\}, \tag{7.3}$$

where $\tau_{\mathrm{c}} \in (0,1), g_k^{\mathrm{c}} = \nabla m_k^{\mathrm{c}}(x_k)$ and $H_k^{\mathrm{c}} = \nabla^2 m_k^{\mathrm{c}}(x_k)$ (we set $\frac{\|g_k^{\mathrm{c}}\|}{\|H_k^{\mathrm{c}}\|} = \infty$ if $\|H_k^{\mathrm{c}}\| = 0$).

Assumption 7.4 *At each iteration k of Algorithm 7.2 at which it is considered, provided $\lambda_k^{\mathrm{e}} = \lambda_{\min}\left(\nabla^2 m_k^{\mathrm{e}}(x_k)\right) < 0$, the model m_k^{e} is minimized by computing an* approximate eigenpoint*, i.e., it exists $\tau_{\mathrm{e}} \in (0,1)$ such that the second-order step satisfies:*

$$m_k^{\mathrm{e}}(x_k) - m_k^{\mathrm{e}}(x_k + s_k^{\mathrm{e}}) \;\geq\; \tau_{\mathrm{e}} \, |\lambda_k^{\mathrm{e}}| \, [\delta_k^{\mathrm{e}}]^2, \tag{7.4}$$

We then provide three general results that lead to convergence. Note that the first one, described in Lemma 7.1, is independent of the quality of the models.

Lemma 7.1 *Let Assumptions 2.2, 7.3 and 7.4 hold. Then there exist positive constants T_{c} and T_{e} depending on initial values and update parameters of the step size, as well as $f(x_0)$ and f_{low}, such that*

$$\sum_{k \in S_+^{\mathrm{c}} \cup U} [\delta_k^{\mathrm{c}}]^2 \;\leq\; T_{\mathrm{c}} < \infty, \tag{7.5}$$

and

$$\sum_{k \in S_+^{\mathrm{e}} \cup U} [\delta_k^{\mathrm{e}}]^3 \;\leq\; T_{\mathrm{e}} < \infty, \tag{7.6}$$

where S_+^{c} (resp. S_+^{e}) is the set of iteration indexes for which δ_k^{c} (resp. δ_k^{e}) possibly increases, and U is the set of unsuccessful iterations (for which both radii are decreased).

Lemma 7.2 *Let Assumptions 6.1, 7.1 and 7.3 hold. Then, provided*

$$\delta_k^{\mathrm{c}} \;<\; \min\left\{ \frac{1}{B_{\mathrm{c}} + \kappa_g^{\mathrm{c}}}, \left[\frac{2\,\kappa_f^{\mathrm{c}}}{\tau_{\mathrm{c}}\,(1 - \eta_1^{\mathrm{c}})} + \kappa_g^{\mathrm{c}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{c}} + \kappa_g^{\mathrm{c}}} \right\} \|\nabla f(x_k)\|, \tag{7.7}$$

the k-th iteration is first-order very successful, i.e., $\rho_k^{\mathrm{c}} \geq \eta_1^{\mathrm{c}}$ and $\|g_k^{\mathrm{c}}\| \geq \eta_2^{\mathrm{c}}\, \delta_k^{\mathrm{c}}$.

Lemma 7.3 *Let Assumptions 6.1, 7.2 and 7.4 hold. Then, provided the true minimum Hessian eigenvalue $\lambda_k = \lambda_{\min}(\nabla^2 f(x_k))$ is negative and*

$$\delta_k^{\mathrm{e}} \;<\; \min\left\{ \left[\frac{\kappa_f^{\mathrm{e}}}{\tau_{\mathrm{e}}\,(1-\eta_2^{\mathrm{e}})} + \kappa_h^{\mathrm{e}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{e}} + \kappa_h^{\mathrm{e}}} \right\} |\lambda_k|, \tag{7.8}$$

the k-th iteration is second-order very successful, i.e., $\rho_k^{\mathrm{e}} \geq \eta_1^{\mathrm{e}}$ and $|\lambda_k^{\mathrm{e}}| \geq \eta_2^{\mathrm{e}}\,\delta_k^{\mathrm{e}}$.

Theorem 7.1 *Under Assumptions 2.2, 6.1, 7.1, 7.2, 7.3 and 7.4,*

$$\liminf_{k\to\infty} \max\left\{ \|\nabla f(x_k)\|, -\lambda_{\min}\left[\nabla^2 f(x_k)\right] \right\} \;=\; 0. \tag{7.9}$$

Proof. Recalling that U is the set of unsuccessful iterations, suppose that it exists $\epsilon > 0$ such that

$$\forall\, k \in U, \quad \max\left\{ \|\nabla f(x_k)\|, -\lambda_{\min}\left[\nabla^2 f(x_k)\right] \right\} > \epsilon. \tag{7.10}$$

Then either there exists a subsequence $K^{\mathrm{c}} \subset U$ for which $\|\nabla f(x_k)\|$ stays above ϵ, or there exists a subsequence $K^{\mathrm{e}} \subset U$ for which $-\lambda_{\min}\left[\nabla^2 f(x_k)\right] > \epsilon$.

Suppose that we are in the first situation. Then, by (7.7), one must have for every $k \in K^{\mathrm{c}}$:

$$\delta_k^{\mathrm{c}} \;\geq\; \min\left\{ \frac{1}{B_{\mathrm{c}} + \kappa_g^{\mathrm{c}}}, \left[\frac{2\,\kappa_f^{\mathrm{c}}}{\tau_{\mathrm{c}}\,(1-\eta_1^{\mathrm{c}})} + \kappa_g^{\mathrm{c}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{c}} + \kappa_g^{\mathrm{c}}} \right\} \epsilon. \tag{7.11}$$

However, from Lemma 7.1, δ_k^{c} goes to zero on unsuccessful iterations U, hence on iterations in K^{c}. Therefore there exists a subsequence $J^{\mathrm{c}} \subset K^{\mathrm{c}}$ such that

$$\lim_{k\in J^{\mathrm{c}}} \|\nabla f(x_k)\| \;=\; 0. \tag{7.12}$$

As a result, for (7.10) to hold, we necessarily have

$$\forall\, k \in J^{\mathrm{c}},\; -\lambda_{\min}\left[\nabla^2 f(x_k)\right] > \epsilon.$$

Using (7.8), we have that

$$\delta_k^{\mathrm{e}} \;\geq\; \min\left\{ \left[\frac{\kappa_f^{\mathrm{e}}}{\tau_{\mathrm{e}}\,(1-\eta_2^{\mathrm{e}})} + \kappa_h^{\mathrm{e}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{e}} + \kappa_h^{\mathrm{e}}} \right\} \epsilon. \tag{7.13}$$

Since $\{\delta_k^{\mathrm{e}}\}_{k\in J^{\mathrm{c}}}$ tends to 0 as a subsequence of a converging sequence, we obtain a contradiction, from which we conclude that there exists a subsequence $I^{\mathrm{c}} \subset J^{\mathrm{c}}$ such that

$$\lim_{k\in I^{\mathrm{c}}} \lambda_{\min}\left[\nabla^2 f(x_k)\right] \;\geq\; 0. \tag{7.14}$$

As a result, for this subsequence, we have

$$\lim_{k\in I^{\mathrm{c}}} \max\left\{ \|\nabla f(x_k)\|, -\lambda_{\min}\left[\nabla^2 f(x_k)\right] \right\} \;=\; 0,$$

which contradicts (7.10).

A symmetric reasoning can be performed in the second situation (involving K^{e}), from which we conclude that (7.9) must hold. □

We now describe the key components of a complexity analysis of Algorithm 7.2. Given $(\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}}) \in (0,1)^2$, we want estimate the cost of reaching an iterate of the method which is an $(\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}})$-approximate stationary point, i.e. at which both

$$\inf_{0 \leq j \leq k} \|\nabla f(x_j)\| \ < \ \epsilon_{\mathrm{c}} \tag{7.15}$$

and

$$\sup_{0 \leq j \leq k} \lambda_{\min}\left(\nabla^2 f(x_j)\right) \ > \ -\epsilon_{\mathrm{e}} \tag{7.16}$$

hold.

By combination of Lemmas 7.2 and 7.3, we can express the properties of the trust-region radii whenever one of the two conditions above is not satisfied.

Lemma 7.4 *Let Assumptions 2.2, 6.1, 7.1, 7.2, 7.3, and 7.4 and suppose that by the k-th iteration an $(\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}})$-approximate critical point has not been reached. Then, for every iterate l between 0 and k such that (7.15) is not satisfied,*

$$\delta_l^{\mathrm{c}} \ \geq \ \gamma_1^{\mathrm{c}} \ \min\left\{ \frac{1}{B_{\mathrm{c}} + \kappa_g^{\mathrm{c}}}, \left[\frac{2\,\kappa_f^{\mathrm{c}}}{\tau_{\mathrm{c}}\,(1-\eta_1^{\mathrm{c}})} + \kappa_g^{\mathrm{c}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{c}} + \kappa_g^{\mathrm{c}}} \right\} \epsilon_{\mathrm{c}}, \tag{7.17}$$

while for every iterate l between 0 and k such that (7.16) does not hold,

$$\delta_l^{\mathrm{e}} \ \geq \ \gamma_2^{\mathrm{e}} \ \min\left\{ \left[\frac{\kappa_f^{\mathrm{e}}}{\tau_{\mathrm{e}}\,(1-\eta_2^{\mathrm{e}})} + \kappa_h^{\mathrm{e}} \right]^{-1}, \frac{1}{\eta_2^{\mathrm{e}} + \kappa_h^{\mathrm{e}}} \right\} \epsilon_{\mathrm{e}}. \tag{7.18}$$

In what follows, we emphasize the dependencies on the model error constants κ_g^{c} and κ_H^{e}, in order to study their dependencies on the dimension n.

Theorem 7.2 *Let Assumptions of Lemma 7.4 hold. Then, the number of iterations needed to attain an $(\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}})$-approximate critical point is*

$$\mathcal{O}\left(\max\left\{ \left[\kappa_g^{\mathrm{c}}\right]^2 \, \epsilon_{\mathrm{c}}^{-2}, [\kappa_h^{\mathrm{e}}]^3 \, \epsilon_{\mathrm{e}}^{-3} \right\} \right), \tag{7.19}$$

where the constant in $\mathcal{O}(\cdot)$ does not depend on ϵ_{c} or ϵ_{e}, but on $f_{\mathrm{low}}, x_0, B_{\mathrm{c}}, , \tau_{\mathrm{c}}, \tau_{\mathrm{e}}, \kappa_f^{\mathrm{c}}$, $\kappa_g^{\mathrm{c}}, \kappa_f^{\mathrm{e}}, \kappa_h^{\mathrm{e}}$ and both versions of algorithmic parameters $\gamma_1, \gamma_2, \eta_1, \eta_2, \delta_0$, indexed by c *or* e.

One sees in the dependences mentioned in Theorem 7.2 that the quality of gradient approximation on the second-order model has no influence on the complexity bound. This confirms the de-coupling of the criteria, although an accurate model gradient is likely to yield better coherence between the model and the objective function.

Interestingly, we observe a bound in $\mathcal{O}\left(\max\left\{\epsilon_{\mathrm{c}}^{-2}, \epsilon_{\mathrm{e}}^{-3}\right\}\right)$, where the first term is always less than the one related to the traditional derivative-based trust-region bound in $\mathcal{O}\left(\max\left\{\epsilon_{\mathrm{c}}^{-2}\,\epsilon_{\mathrm{e}}^{-1}, \epsilon_{\mathrm{e}}^{-3}\right\}\right)$ for derivative-based trust-region schemes [25]. Such a result was to be expected given the decrease requirements on our method. As discussed at the end of Section 6.3, as long as an approximate stationary point has not been reached, one would have

$$f(x_k) - f(x_{k+1}) \;\geq\; \mathcal{O}\left(\min\left\{\epsilon_{\mathrm{c}}\,\delta_k, \epsilon_{\mathrm{e}}\,\delta_k^2\right\}\right) \quad \text{and} \quad \delta_k \;\geq\; \mathcal{O}\left(\min\left\{\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}}\right\}\right) \tag{7.20}$$

on iterations of Algorithm 7.1 for which $\delta_{k+1} = \min\left\{\gamma_2\,\delta_k, \delta_{\max}\right\}$. In the case of Algorithm 7.2, those conditions involving a minimum are replaced by separated ones acting on different subsequences of iterations. Still, the convergence of the trust-region radii on unsuccessful iterations allows for the derivation of a global complexity result.

Note that in Algorithm 7.2, $\frac{(n+1)(n+2)}{2} + 1$ function values are necessary in the worst case (which will be considered in our numerical experiments) to build a fully quadratic model, while $2\,n + 1$ evaluations are required for constructing a fully linear model, again in the worst case. As a result, the number of evaluations is bounded as follows.

Theorem 7.3 *Let the assumptions of Lemma 7.4 hold. Then, the number of iterations needed to reach an $(\epsilon_{\mathrm{c}}, \epsilon_{\mathrm{e}})$-approximate critical point is*

$$\mathcal{O}\left(n^2\,\max\left\{\left\lceil\kappa_g^{\mathrm{c}}\right\rceil^2\,\epsilon_{\mathrm{c}}^{-2}, \left\lceil\kappa_h^{\mathrm{e}}\right\rceil^3\,\epsilon_{\mathrm{e}}^{-3}\right\}\right), \tag{7.21}$$

where the constant in $\mathcal{O}(\cdot)$ does not depend on ϵ_{c} or ϵ_{e}, but on $f_{\mathrm{low}}, x_0, B_{\mathrm{c}}, \tau_{\mathrm{c}}, \tau_{\mathrm{e}}, \kappa_f^{\mathrm{c}}$, $\kappa_g^{\mathrm{c}}, \kappa_f^{\mathrm{e}}, \kappa_h^{\mathrm{e}}$ and both versions of algorithmic parameters $\gamma_1, \gamma_2, \eta_1, \eta_2, \delta_0$, indexed by c *or* e*.*

In the case of a model computed through polynomial interpolation, one can show [35] that

$$\begin{aligned} \kappa_g^{\mathrm{c}} &= \mathcal{O}(\kappa_f^{\mathrm{c}}) &&= \mathcal{O}(\sqrt{n}) \\ \kappa_h^{\mathrm{e}} &= \mathcal{O}(\kappa_f^{\mathrm{e}}) &&= \mathcal{O}(n), \end{aligned}$$

We can then replace these constants in the bounds, which yields a dependence in n^5 with respect to the dimension. Such dependencies have already been observed in globally convergent second-order derivative-free methods [66, 73]. For the particular case of the polynomial interpolation, we have more precisely

$$\mathcal{O}\left(\max\left\{n^3\,\epsilon_{\mathrm{c}}^{-2}, n^5\,\epsilon_{\mathrm{e}}^{-3}\right\}\right).$$

7.3 A de-coupled direct search with improved complexity results

In this section, we propose a direct-search method based on Algorithm 6.1 with a decoupling strategy. The essential characteristic of the new method is that it relies on two

different step sizes in order to account for the first and second-order optimality criteria. This is particularly interesting if those criteria differ in magnitude; in that situation, a single step size could only be appropriately scaled with respect to one of the two criteria at a time. Proceeding with two step sizes allows to tackle both aspects in a separate fashion.

7.3.1 Algorithmic process and convergence properties

Algorithm 7.3 describes how the minimization process is carried on. As in Algorithm 6.1, no search step is performed and the directions of interest are unitary.

A key feature of Algorithm 7.3 is the update rules on the step sizes. One sees that whenever at a successful step, the algorithm will (possibly) increase the step size that has produced decrease while leaving the other step size unchanged for the next iteration. This process allows the method to cope with different orders of magnitude between the first and second-order aspects, a common issue in nonconvex optimization based upon directional decrease [12].

The forcing functions will be considered to satisfy the following assumption. We will require different decreases for the first and the second-order polling.

Assumption 7.5 *The forcing function $\rho_1, \rho_2 : \mathbb{R}^+ \to \mathbb{R}^+$ satisfies the following properties:*

i) the functions are non-decreasing,

ii) $\rho_1(\alpha) = o(\alpha)$ when $\alpha \to 0^+$,

iii) $\rho_2(\beta) = o(\beta^2)$ when $\beta \to 0^+$.

For the rest of the section, S^1 will denote the successful iterations for which a direction satisfying (7.22) was obtained. Similarly, S^2 will be the set of indexes at which a direction satisfying (7.23) was found. The set of unsuccessful iterations will be indicated by U.

Lemma 7.5 *Under Assumption 2.2, one has*

$$\sum_{k \in S^1 \cup U} \rho_1(\alpha_k) \; < \; \infty \tag{7.25}$$

which implies that $\{\alpha_k\}_{k \in S^1 \cup U}$ goes to 0, and

$$\sum_{k \in S^2 \cup U} \rho_2(\beta_k) \; < \; \infty, \tag{7.26}$$

so $\{\beta_k\}_{k \in S^2 \cup U}$ also goes to zero.

Proof. For both step sizes, the proof follows the process of that of Lemma 3.7, only with S^1 (resp. S^2) replacing the successful iterations for the series of α_k (resp. β_k). □

Algorithm 7.3: De-coupled Step sizes Direct-Search method (DSDS)

Choose an initial point $x_0 \in \mathbb{R}^n$, as well as $0 < \theta_\alpha < 1 \leq \gamma_\alpha$, $0 < \theta_\beta < 1 \leq \gamma_\beta$, $0 < \alpha_0 < \alpha_{\max}$, $0 < \beta_0 < \beta_{\max}$. Define two *forcing functions* $\rho_1, \rho_2 : \mathbb{R}^+ \to \mathbb{R}^+$.

for $k = 0, 1, 2, \ldots$ **do**

1 - First-Order Poll Step

Generate a Positive Spanning Set D_k. If there exists $d \in D_k$ such that

$$f(x_k + \alpha_k\, d) \;<\; f(x_k) - \rho_1(\alpha_k), \tag{7.22}$$

declare iteration k as *first-order successful* with $d_k = d$ and go to the update step, otherwise go to the second-order poll step.

2 - Second-Order Poll Step

(a) Generate a (linear) basis B_k; if it exists $d \in [B_k \text{ -}B_k]$ such that

$$f(x_k + \beta_k\, d) \;<\; f(x_k) - \rho_2(\beta_k), \tag{7.23}$$

declare iteration k as *second-order successful* with $d_k = d$ and go the update step.

(b) Otherwise, let $d_1, \cdots, d_n$ be an order on the directions in B_k. If it exists $d \in \{d_i + d_j, 1 \leq i < j \leq n\}$ such that (7.23) holds, then declare the iteration successful with $d_k = d$, and go to the update step.

(c) Otherwise, define the matrix H_k by setting for all $(i, j) \in \{1, \ldots, n\}^2, i < j$:

$$\begin{aligned} (H_k)_{ii} &= \tfrac{f(x_k+\beta_k\, d_i)-2\, f(x_k)+f(x_k-\beta_k\, d_i)}{\beta_k^2}, \\ (H_k)_{ij} &= \tfrac{f(x_k+\beta_k\, d_i+\beta_k\, d_j)-f(x_k+\beta_k\, d_i)-f(x_k+\beta_k\, d_j)+f(x_k)}{\beta_k^2}. \end{aligned} \tag{7.24}$$

(d) Compute a unitary eigenvector v_k associated with $\lambda_{\min}(H_k)$. If v_k or -v_k satisfies (7.23), declare the iteration as *second-order successful* with d_k equal to v_k or -v_k, otherwise declare it *unsuccessful*.

3 - Update Step

If the iteration is first-order successful, set $x_{k+1} = x_k + \alpha_k\, d_k$, $\alpha_{k+1} = \min\{\gamma_\alpha\, \alpha_k, \alpha_{\max}\}$ and $\beta_{k+1} = \beta_k$.

If the iteration is second-order successful, set $x_{k+1} = x_k + \beta_k\, d_k$, $\beta_{k+1} = \min\{\gamma_\beta\, \beta_k, \beta_{\max}\}$ and $\alpha_{k+1} = \alpha_k$.

Otherwise, set $x_{k+1} = x_k$, $\alpha_{k+1} = \theta_\alpha\, \alpha_k$, and $\beta_{k+1} = \theta_\beta\, \beta_k$.

end

Note that this result is weaker with respect to each of the step sizes compared to the previous direct-search methods we studied, but it still suffices to obtain a liminf-type convergence result as well as a worst-case complexity bound.

We now state the equivalent of Lemma 6.3 for Algorithm 7.3, which one can prove by following the same reasoning.

Lemma 7.6 *Let k be the index of an unsuccessful iteration of Algorithm 7.3 such that $\mathrm{cm}(D_k) \geq \kappa_g$ and $\sigma_{\min}(B_k)^2 \geq \sigma$ for $(\kappa_g, \sigma) \in (0,1)^2$. Under Assumption 6.1,*

$$\|\nabla f(x_k)\| \;\leq\; \kappa_g^{-1}\left(\frac{\rho_1(\alpha_k)}{\alpha_k} + \frac{\nu_g}{2}\alpha_k\right) \tag{7.27}$$

is satisfied, and, if $\lambda_k = \lambda_{\min}\left(\nabla^2 f(x_k)\right) < 0$,

$$\lambda_k \;\geq\; -\sigma^{-1}\left(\frac{2\,\rho_2(\beta_k)}{\beta_k^2} + (10\,n+1)\frac{\nu_H}{3}\beta_k\right) \tag{7.28}$$

also holds.

Lemma 7.6 implies that every criterion is related to one step size, that will serve to its estimation. Since both subsequences of step sizes corresponding to unsuccessful iterations go to zero by Lemma 7.5, we can establish convergence of the method under the following assumptions on the sets D_k and B_k.

Assumption 7.6 *It exists $\kappa_g \in (0,1)$ such that for every k, the first-order polling set D_k is a κ_g-descent set.*

Assumption 7.7 *It exists $\sigma \in (0,1]$ such that at any iteration k, the set B_k (viewed as an n-by-n matrix) satisfies $\sigma_{\min}(B_k)^2 \;\geq\; \sigma$.*

Theorem 7.4 *Under Assumptions 2.2, 6.1, 7.5, 7.6 and 7.7, one has*

$$\liminf_{k\to\infty} \max\left\{\|\nabla f(x_k)\|, -\lambda_{\min}\left(\nabla^2 f(x_k)\right)\right\} \;=\; 0. \tag{7.29}$$

Algorithm 7.3 is thus second-order globally convergent.

7.3.2 Complexity analysis

The reasoning of Section 6.3.3 can be adapted to derive a worst-case complexity analysis of Algorithm 7.3. We recall that our objective is to bound the number of iterations needed to achieve:

$$\inf_{0\leq l\leq k} \|\nabla f(x_l)\| < \epsilon_g \quad \text{and} \quad \sup_{0\leq l\leq k} \lambda_{\min}\left(\nabla^2 f(x_l)\right) > -\epsilon_H. \tag{7.30}$$

Because of the de-coupling property of the method, we can establish separate bounds on the number of first-order successful and second-order successful iterations. This results in a worst-case estimate that possibly improves the one presented in Section 6.3.3.

Such a difference is due to the disjunction between first-order and second-order sufficient decreases, that involve not only two step sizes but also two forcing functions. The independence between the treatment of both optimality criteria is made clear in Proposition 7.1.

Proposition 7.1 *Suppose that the assumptions of Theorem 7.4 hold. Given ϵ_g and ϵ_H in $(0,1)$, let k_0 be the index of the first unsuccessful iteration and assume that (7.30) does not hold, and let l_1 be the first index such that (7.30) is satisfied at iteration l_1+1.*

Then, the number of successful iterations between k_0 and l_1 satisfies

$$|S_{l_1}(k_0)| \;=\; \max\left\{|S^1(k_0)|, |S^2(k_0)|\right\}, \tag{7.31}$$

where $S^1(k_0)$ (resp. $S^2(k_0)$) denotes the set of first-order (resp. second-order) successful iterations k for which $\inf_{0\leq l\leq k}\|\nabla f(x_l)\| \geq \epsilon_g$ (resp. $\sup_{0\leq l\leq k}\lambda_{\min}(\nabla^2 f(x_l)) \leq -\epsilon_H$).

Theorem 7.5 *Suppose that the assumptions of Theorem 7.4 hold. Assume that Algorithm 7.3 is applied with $\rho_1(\alpha) = \frac{c_1}{2}\,\alpha^q$, $\rho_2(\beta) = \frac{c_2}{6}\,\beta^r$ with $c_1, c_2 > 0$, $q > 1$ and $r > 2$. Define $\hat{q} = \frac{q}{\min\{q-1,1\}}$ and $\hat{r} = \frac{r}{\min\{r-2,1\}}$.*

Given $\epsilon_g, \epsilon_H \in (0,1)$, let k_0 be the index of the first unsuccessful iteration and assume that (7.30) does not hold, and let l_1 be the first index such that (7.30) is satisfied at iteration l_1+1. Then the number of successful iterations between k_0 and l_1, denoted by $|S_{l_1}(k_0)|$, is bounded as follows:

$$|S_{l_1}(k_0)| \;\leq\; \left\lceil \left(\frac{f(x_{l_0}) - f_{\text{low}}}{L_s}\right) \max\left(\kappa_g^{-\hat{q}}\,\epsilon_g^{-\hat{q}}, (\sigma/n)^{-\hat{r}}\,\epsilon_H^{-\hat{r}}\right)\right\rceil, \tag{7.32}$$

where

$$L_s = \min\left\{\frac{c_1\,\theta_\alpha^{\hat{q}}}{2}, \frac{c_2\,\theta_\beta^{\hat{r}}}{6}\right\} \min\left\{1, L_1^{-\hat{q}}, L_2^{-\hat{r}}\right\}, L_1 = \frac{c_1+\nu_g}{2}, \text{ and } L_2 = \frac{c_2+11\,\nu_H}{3}.$$

Proof. Thanks to Proposition 7.1, we consider the criteria in an independent fashion.

Assume that $S^1(k_0)$ is not empty (otherwise the upcoming result is trivial), and denote by l_{11} the maximum index in $S^1(k_0)$. For every $l \in U$ such that $k_0 \leq l < l_{11}$, we then know that

$$\|\nabla f(x_l)\| \;\geq\; \inf_{0\leq j\leq l}\|\nabla f(x_j)\| \;\geq \epsilon_g.$$

Thus, by Lemma 7.6,

$$\epsilon_g \leq \|\nabla f(x_l)\| \;\leq\; \frac{1}{\kappa_g}\left[\frac{c_1}{2}\,\alpha_l^{q-1} + \frac{\nu_g}{2}\alpha_l\right]. \tag{7.33}$$

By distinguishing the cases $\alpha_l \geq 1$ and $\alpha_l < 1$, we arrive at

$$\alpha_l \;\geq\; \min\left(1, L_1^{-\frac{\hat{q}}{q}}\right)\,\kappa_g^{\frac{\hat{q}}{q}}\,\epsilon_g^{\frac{\hat{q}}{q}}. \tag{7.34}$$

Considering now the iterations in $S^1(k_0)$, one can backtrack to the previous index in U (which is always greater than or equal to k_0). As a result, the bound on the first-order step size for every index $l \in S^1(k_0)$ between k_0 and l_{11} is

$$\alpha_l \;\geq\; \theta_\alpha \, \min\left(1, L_1^{-\frac{\hat{q}}{q}}\right) \, \kappa_g^{\frac{\hat{q}}{q}} \, \epsilon_g^{\frac{\hat{q}}{q}}. \tag{7.35}$$

If we look at the function value decreases, we have that

$$\begin{aligned}
\sum_{l \in S^1(k_0)} f(x_l) - f(x_{l+1}) \;&\geq\; \sum_{l \in S^1(k_0)} \rho_1(\alpha_l) \\
&\geq\; \frac{c_1}{2} \sum_{l \in S^1(k_0)} \alpha_l^q \\
&\geq\; \left|S^1(k_0)\right| \frac{c_1}{2} \, \theta_\alpha^q \, \min\left(1, L_1^{-\hat{q}}\right) \, \kappa_g^{\hat{q}} \, \epsilon_g^{\hat{q}}.
\end{aligned}$$

Since

$$\sum_{l \in S^1(k_0)} f(x_l) - f(x_{l+1}) \;\leq\; \sum_{l \in S_{l_1}(k_0)} f(x_l) - f(x_{l+1}) \;\leq\; f(x_{k_0}) - f(x_{l_1+1}) \;\leq\; f_0 - f_{\text{low}},$$

one obtains the first-order bound

$$\left|S^1(k_0)\right| \;\leq\; \frac{2(f_0 - f_{\text{low}})}{c_1 \, \theta_\alpha^q \, \min\left(1, L_1^{-\hat{q}}\right)} \, \kappa_g^{-\hat{q}} \, \epsilon_g^{-\hat{q}}. \tag{7.36}$$

A similar study for the set $S^2(k_0)$ leads to

$$\left|S^2(k_0)\right| \;\leq\; \frac{6(f_0 - f_{\text{low}})}{c_2 \, \theta_\beta^r \, \min\left(1, L_2^{-\hat{r}}\right)} \, (\sigma/n)^{-\hat{r}} \, \epsilon_H^{-\hat{r}}. \tag{7.37}$$

We arrive at our final result by applying Proposition 7.1. □

Theorem 7.6 *Let the assumptions of Theorem 7.5 hold. Then, with the same definitions for k_0 and l_1, the number of unsuccessful iterations between k_0 and l_1 is at most $|U_{l_1}(k_0)|$, where*

$$|U_{l_1}(k_0)| \;\leq\; \frac{1}{2} \left\lceil L_3 \, |S_{l_1}(k_0)| + L_4 - (\log_{\theta_\alpha} e) \, \kappa_g^{-\frac{\hat{q}}{q}} \, \epsilon_g^{-\frac{\hat{q}}{q}} - (\log_{\theta_\beta} e) \, (\sigma/n)^{-\frac{\hat{r}}{r}} \, \epsilon_H^{-\frac{\hat{r}}{r}} \right\rceil \tag{7.38}$$

with

$$L_3 = \max\left\{-\log_{\theta_\alpha} \gamma_\alpha, -\log_{\theta_\beta} \gamma_\beta\right\}$$

and

$$L_4 = \log_{\theta_\alpha}\left(\frac{\theta_\alpha \, \min\{1, L_1^{-\frac{\hat{q}}{q}}\} \, e}{\alpha_{k_0}}\right) + \log_{\theta_\beta}\left(\frac{\theta_\beta \, \min(1, L_2^{-\frac{\hat{r}}{r}}) \, e}{\beta_{k_0}}\right).$$

Proof. We base ourselves on the proof of Theorem 6.4. Given the updating rules on the first-order step size α_k, we have that

$$\alpha_{l_1} \;\leq\; \alpha_{k_0}\, \theta_\alpha^{|U_{l_1}(k_0)|}\, \gamma_\alpha^{|S^1(k_0)|}, \tag{7.39}$$

thus, by reasoning as in Theorem 6.4, one can arrive at

$$|U_{l_1}(k_0)| \;\leq\; -\log_{\theta_\alpha} \gamma_\alpha \left|S^1(k_0)\right| + \log_{\theta_\alpha} \left(\frac{\theta_\alpha \, \min(1, L_1^{-\frac{\hat{q}}{q}})\, e}{\alpha_{k_0}} \right) - \log_{\theta_\alpha} e\, (\kappa_g\, \epsilon_g)^{-\frac{\hat{q}}{q}}. \tag{7.40}$$

Similarly, one has

$$|U_{l_1}(k_0)| \;\leq\; -\log_{\theta_\beta} \gamma_\beta \left|S^2(k_0)\right| + \log_{\theta_\beta} \left(\frac{\theta_\beta \, \min(1, L_2^{-\frac{\hat{r}}{r}})\, e}{\beta_{k_0}} \right) - \log_{\theta_\beta} e\, \left(\sigma\, n^{-1}\, \epsilon_H\right)^{-\frac{\hat{r}}{r}}. \tag{7.41}$$

Summing (7.36) and (7.37) yields

$$\begin{aligned} 2\, |U_{l_1}(k_0)| \;\;\leq\;\; & L_3\, |S_{l_1}(k_0)| + L_4 - (\log_{\theta_\alpha} e)\, \kappa_g^{-\frac{\hat{q}}{q}}\, \epsilon_g^{-\frac{\hat{q}}{q}} \\ & -(\log_{\theta_\beta} e)\, (\sigma/n)^{-\frac{\hat{r}}{r}}\, \epsilon_H^{-\frac{\hat{r}}{r}}, \end{aligned}$$

hence the result. □

Since the index of the first unsuccessful iteration can be shown to be at most

$$\left\lceil (f(x_0) - f_{\text{low}})\, \max\left\{ \frac{2}{c_1\, \alpha_0^p}\, \kappa_g^{-\hat{q}}\, \epsilon_g^{-\hat{q}}, \frac{6}{c_2\, \beta_0^r}\, (\sigma/n)^{-\hat{r}}\, \epsilon_H^{-\hat{r}} \right\} \right\rceil,$$

we obtain the following complexity results for Algorithm 7.3.

Theorem 7.7 *Let the assumptions of Theorem 7.5 hold. The number of iterations needed by Algorithm 7.3 to satisfy* (7.30) *is at most*

$$\mathcal{O}\left(\max\left(\kappa_g^{-\hat{q}}\, \epsilon_g^{-\hat{q}}, (\sigma/n)^{-\hat{r}}\, \epsilon_H^{-\hat{r}} \right) \right), \tag{7.42}$$

where the constant in $\mathcal{O}(\cdot)$ depends on $\nu_g, \nu_H, \alpha_0, \beta_0, f(x_0), f_{\text{low}}, c_1, c_2, \gamma_\alpha, \theta_\alpha, \gamma_\beta, \theta_\beta, q$ and r. In addition, the number of function evaluations needed by Algorithm 6.1 is at most

$$\mathcal{O}\left(m\, \max\left(\kappa_g^{-\hat{q}}\, \epsilon_g^{-\hat{q}}, (\sigma/n)^{\hat{r}}\, \epsilon_H^{-\hat{r}} \right) \right), \tag{7.43}$$

with m is the maximum number of function evaluations performed in any iteration, and the constant in $\mathcal{O}(\cdot)$ is the same as in (7.42).

It comes out from the bounds in Theorem 7.7 that Algorithm DSDS presents a better complexity in terms of first-order optimality, without worsening the number of function evaluations in terms of dependence on n. For instance, by using $D_k = D_\oplus$ at every

iteration and extract B_k directly from D_k, the maximum number if evaluations one has to perform in a single iteration of Algorithm 7.3 is

$$m \;=\; 2\,n + 2\,n + \frac{n^2 - n}{2} + 2 \;=\; \frac{n^2 + 7\,n + 4}{2},$$

while if we set $D_k = V_{n+1}$ and use $B_k = I$ (the canonical basis), this maximum number of evaluations per iteration becomes

$$m \;=\; n + 1 + 2\,n + \frac{n^2 - n}{2} + 2 \;=\; \frac{n^2 + 5\,n + 6}{2}$$

As a result, $m = \mathcal{O}(n^2)$, so its dependence on n does not worsen compared to Algorithm 6.1, even though the total number of evaluations does. The lowest $\hat{q}$ and $\hat{r}$ being obtained by choosing $q = 2$ and $r = 3$, we arrive at a bound in

$$\mathcal{O}\left(\max\left\{n^3\,\epsilon_g^{-2}, n^5\,\epsilon_H^{-3}\right\}\right) \tag{7.44}$$

for the amount of function evaluations. One thus sees that we recover the results from Section 7.2.2, as the improvement on the power of the tolerance ϵ_g also affects the dependence on n with respect to the first-order criterion. Nevertheless, note that for ϵ_H sufficiently smaller than ϵ_g, the second part of the bound is the dominant one, and we obtain a bound in $\mathcal{O}\left(n^5\,\epsilon_H^{-3}\right)$.

7.4 Towards randomization

An interesting possibility offered by the de-coupling techniques is the introduction of randomness with respect to one single aspect. Indeed, since the treatments of first and second-order aspects are done in a dissociated way, it is possible to consider one of them from a probabilistic perspective while the other one is treated deterministically, as in the previous sections. We describe below the results that can be obtained by introducing probabilistic first-order properties.

7.4.1 De-coupled trust regions with probabilistic first-order models

Considering Algorithm 7.2 applied to a nonconvex objective function, we now assume that the first-order models satisfy Assumption 7.8.

Assumption 7.8 *The model sequence is a $(p, \kappa_g^c, \kappa_f^c)$-fully linear sequence, with $p \geq p_0$ and*

$$p_0 \;=\; \frac{\ln(\gamma_1^c)}{\ln\left(\gamma_1^c/\gamma_2^c\right)}.$$

Meanwhile, the second-order will be required to satisfy the *deterministic* fully quadratic property of Assumption 7.2. Note that the second-order models will be considered as random as the whole algorithmic process (in particular, the iterates and the trust-region radii) turns into a random one. As we have seen, this does not prevent from ensuring deterministic properties.

Theorem 7.8 *Let Assumptions 2.2 and 6.1 hold. Consider the application of Algorithm 7.2 based on probabilistically fully linear models satisfying Assumption 7.8 and deterministically fully quadratic models satisfying Assumption 7.2. Suppose that those models are approximately minimized so that Assumptions 7.3 and 7.4 are valid for all realizations of the method.*

Then, the random sequence of iterates produced by the algorithm satisfies

$$\mathbb{P}\left(\max\left\{\left\|\nabla f(X_k)\right\|, -\lambda_{\min}\left[\nabla^2 f(X_k)\right]\right\} = 0\right) = 1.$$

A probabilistic complexity analysis of the proposed scheme can also be derived, yielding a worst-case complexity bound in $\mathcal{O}\left(n^2 \max\left\{\left[\kappa_g^{\mathrm{c}}\right]^2 \epsilon_{\mathrm{c}}^{-2}, [\kappa_h^{\mathrm{e}}]^3 \epsilon_{\mathrm{e}}^{-3}\right\}\right)$ in terms of function evaluations, holding with overwhelming probability and matching the deterministic ones.

7.4.2 De-coupled direct search based on probabilistic first-order descent

We now consider Algorithm 7.3 and assume that the first-order polling sets satisfy a probabilistic descent property: as in the previous section, we target almost-sure second-order convergence.

Assumption 7.9 *The polling set sequence of Algorithm 7.3 is made of sets of m random independent vectors, and form a (p, κ_g)-descent sequence with $p \geq p_0$, where*

$$p_0 = \frac{\ln \theta_\alpha}{\ln\left(\gamma_\alpha^{-1}\theta_\alpha\right)}. \tag{7.45}$$

The bases B_k will however be treated as deterministic, and required to satisfy Assumption 7.7. Note that contrary to the trust-region case, the sets B_k do not become random due to the first-order polling sets (however the iterates and step sizes do).

The convergence result one can obtain under this setting is stated in Theorem 7.9.

Theorem 7.9 *Let Assumptions 2.2, 6.1 and 7.5 hold. Consider the application of Algorithm 7.3 based on random (first-order) polling sets satisfying Assumption 7.9 and deterministic (second-order) linear bases satisfying Assumption 7.7.*

Then, the random sequence of iterates produced by the algorithm satisfies

$$\mathbb{P}\left(\max\left\{\left\|\nabla f(X_k)\right\|, -\lambda_{\min}\left[\nabla^2 f(X_k)\right]\right\} = 0\right) = 1.$$

It is also possible to derive a probabilistic worst-case complexity bound under the assumptions of Theorem 7.9. Following the reasoning of Section 3.4, one can show that with overwhelming probability, the number of function evaluations is in

$$\mathcal{O}\left(m \max\left\{\kappa_g^{-2} \epsilon_{\mathrm{c}}^{-2}, [n/\sigma]^3 \epsilon_{\mathrm{e}}^{-3}\right\}\right),$$

with m being the maximum number of function evaluations performed in one iteration. Although it remains of order of n^2 in the worst case, it can be reduced when less than $n+1$ directions are used to build D_k. For instance, with 2 directions uniformly distributed in the sphere, one would obtain

$$m = 2 + 2\,n + \frac{n^2 - n}{2} + 2 = \frac{n^2 + 3\,n + 8}{2},$$

which for $n \geq 2$ is always less than the maximum number of iterations performed in the deterministic case studied in Section 7.3.

7.5 Numerical study of the de-coupled approach

In this section, we investigate the numerical impact of our de-coupling strategy. It is shown to yield improvement in the trust-region case, while offering new insights on deterministic and probabilistic properties in direct-search algorithms.

7.5.1 Preliminary experiments on the de-coupled trust-region method

We tested an implementation of the DESTRESS algorithm on a benchmark consisting of 55 `CUTEst` [65] problems among the nonconvex benchmark given in Appendix B. Comparison was made with the basic second-order convergent trust-region relying on quadratic interpolation models described in Algorithm 7.1. To highlight the pathological behavior of those two methods, a new poised sample set was generated at every iteration. For Algorithm 7.1, the models were computed by interpolation so as to satisfy a fully quadratic property [35]. In the case of Algorithm 7.2 (DESTRESS), both models were quadratic polynomials. The models $\{m_k^{\mathrm{c}}\}$ were computed by linear interpolation (in which case a diagonal Hessian based on symmetric polling was used), while for the models $\{m_k^{\mathrm{e}}\}$ quadratic polynomial interpolation was used. In Algorithm 7.1, we used quadratic polynomial interpolation models in order to guarantee second-order convergence of the method. Note that the remaining parameters of the trust-region subproblems and updates were set to be identical (for instance, we set $\eta_1 = \eta_1^{\mathrm{c}} = \eta_1^{\mathrm{e}}$).

For these methods, we gave a budget of 500 $\left(\frac{(n+1)(n+2)}{2} + 2n + 1\right)$ evaluations (i.e. a budget for computing both a first-order and a second-order model for 500 iterations). The stopping criterion was

$$f(x_k) - f_{best} \;<\; \epsilon_f \left(f_0 - f_{best}\right),$$

where f_{best} was the best value obtained by those solvers with the given budget of evaluations and ϵ_f was a given tolerance. In addition to the stopping criterion, we stopped the method if the budget was exhausted or if the trust-region radius (one of the two in case of Algorithm 7.2) dropped below 10^{-8}.

Figures 7.1 to 7.3 present the results for various values of the tolerance ϵ_f. In the derivative-free setting, Algorithm 7.2 outperforms a standard approach method, essentially because of its ability to save function evaluations if the first-order trust-region

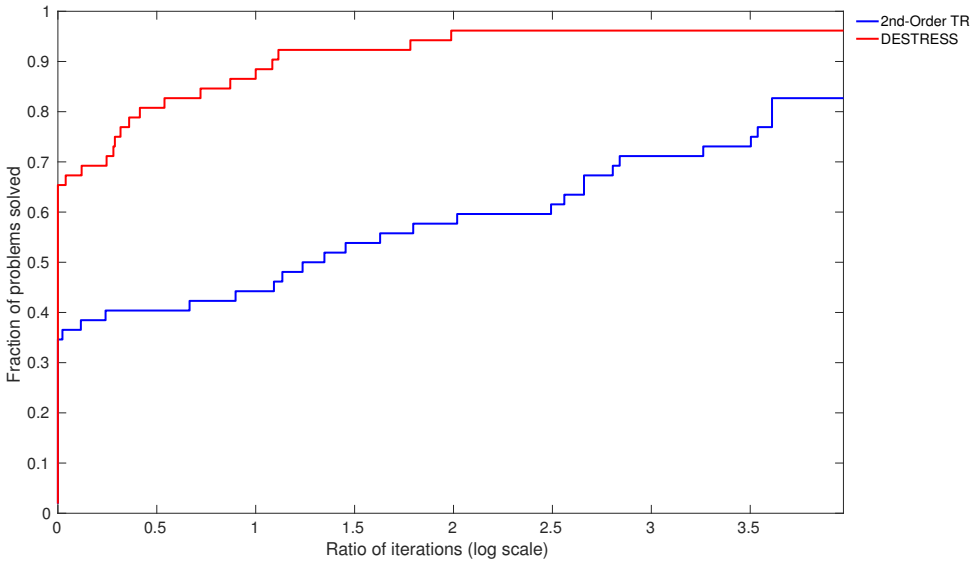


Figure 7.1: Performance of the derivative-free trust-region methods for $\epsilon_f = 10^{-3}$.

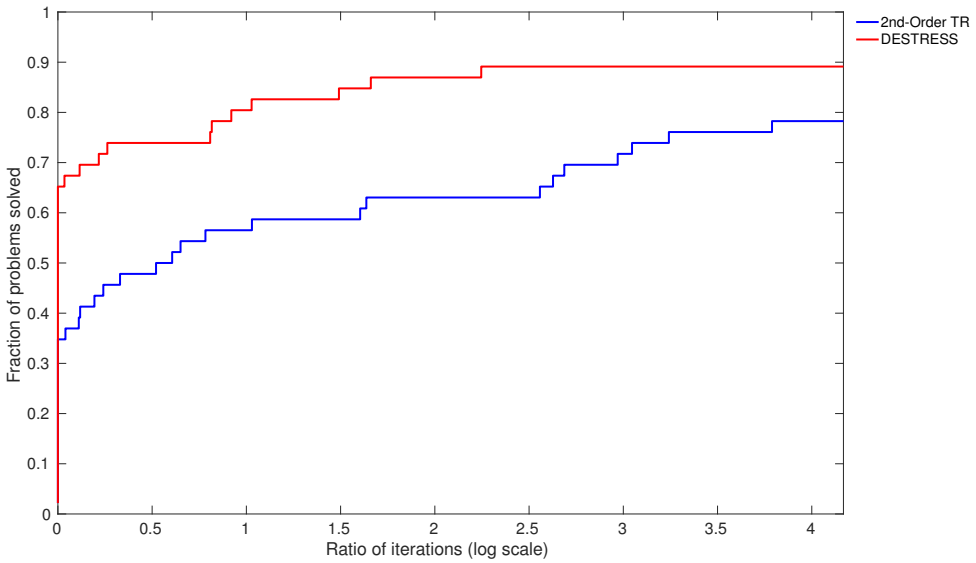


Figure 7.2: Performance of the derivative-free trust-region methods for $\epsilon_f = 10^{-6}$.

step is successful. It also appears that a separate second-order step is able to provide more decrease in the problem, thus improving the general robustness of the method.

Note that the results were similar when the classical decrease ratios were replaced

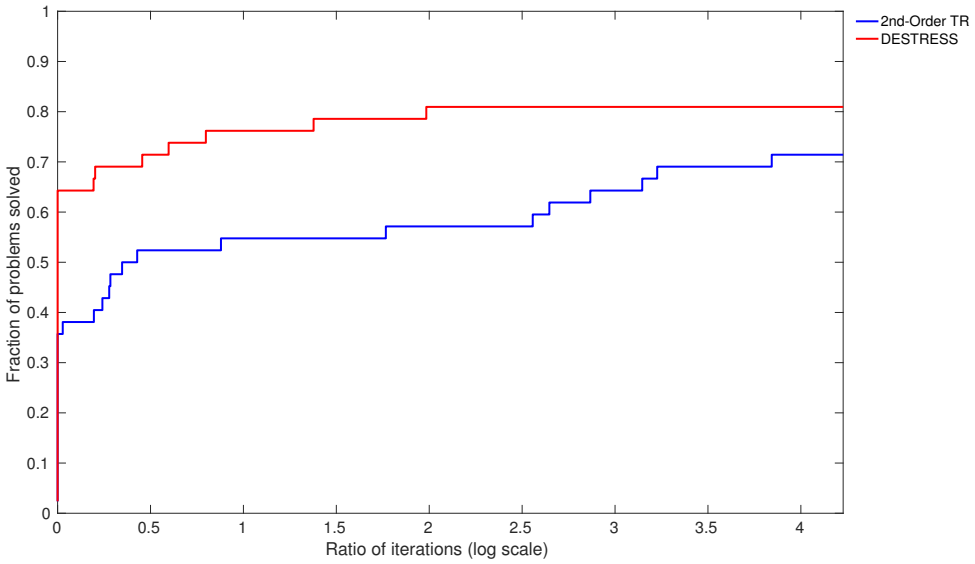


Figure 7.3: Performance of the derivative-free trust-region methods for $\epsilon_f = 10^{-9}$.

by

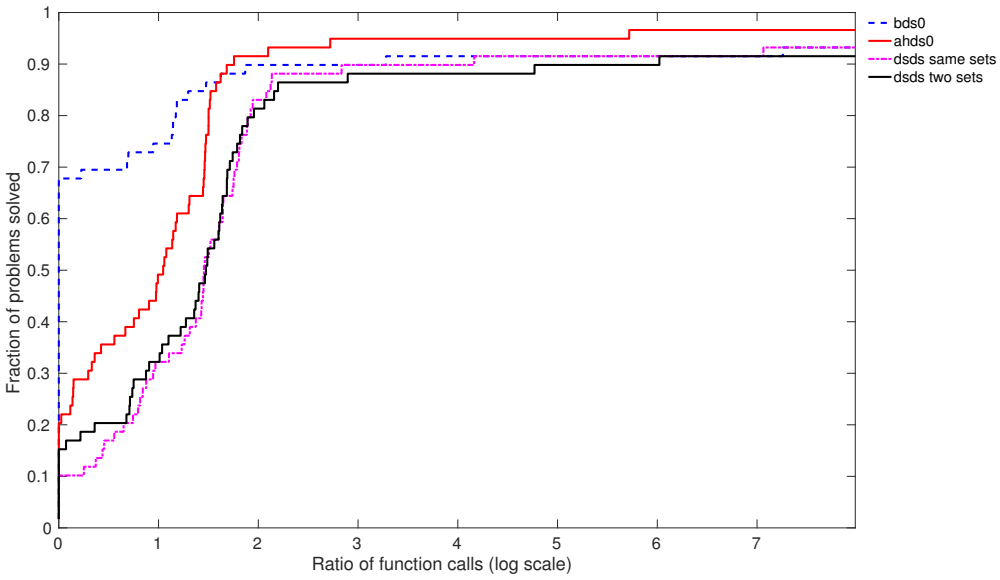
$$\rho_k^{\mathrm{c}} = \frac{f(x_k) - f(x_k + s_k^{\mathrm{c}}) - c_{\mathrm{c}}\,[\delta_k^{\mathrm{c}}]^2}{m_k^{\mathrm{c}}(x_k) - m_k^{\mathrm{c}}(x_k + s_k^{\mathrm{c}})}, \quad \rho_k^{\mathrm{e}} = \frac{f(x_k) - f(x_k + s_k^{\mathrm{e}}) - c_{\mathrm{e}}\,[\delta_k^{\mathrm{c}}]^3}{m_k^{\mathrm{e}}(x_k) - m_k^{\mathrm{e}}(x_k + s_k^{\mathrm{e}})}, \tag{7.46}$$

which are the decoupled equivalents of those presented in [60] for deriving worst-case complexity bounds in the composite and nonsmooth settings.

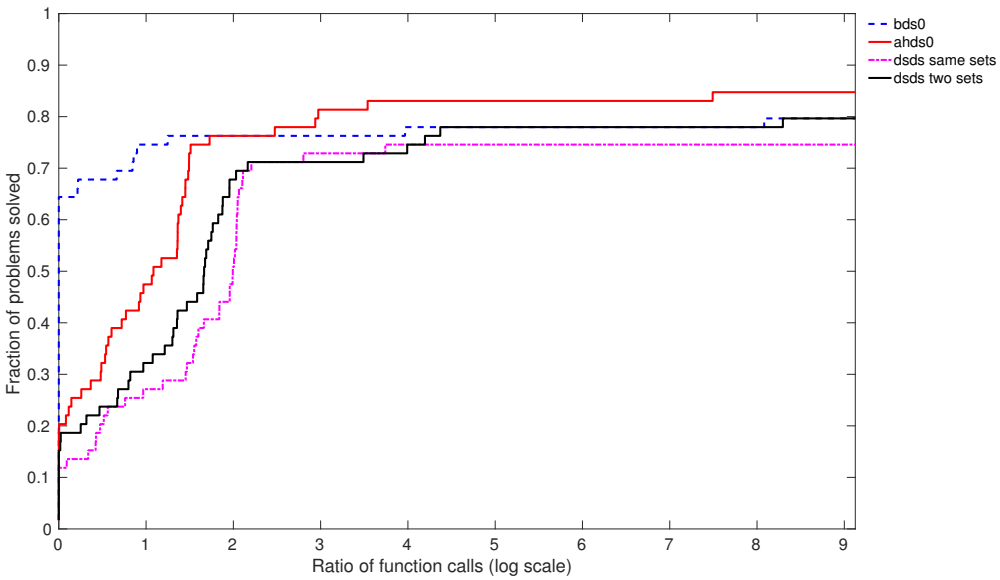
7.5.2 Numerical study of the de-coupled direct-search framework

We hereby present a more extensive experimentation with the de-coupled framework of Algorithm 7.3. Although similar in spirit to Algorithm 6.1 regarding the treatment of second-order aspects, it handles the first-order ones differently. Having established that this is beneficial for the complexity analysis, we now aim at studying the impact of such a technique on the practical performance.

Our experimental setting is quite similar to that of the previous chapter, with the same benchmark of 60 `CUTEst` problems as in Section 6.4. We focused on the best variants of Algorithms 2.1 and 6.1, corresponding to the polling choice 0 ($D_\oplus$). For comparison, we selected two variants on the DSDS framework, respectively based on considering the same polling sets for both the first and the second-order polling. In the first case, we used $D_k = D_\oplus$, while in the second case, the chosen PSS was $D_k = Q\,D_\oplus$ (i.e., also a symmetric set). For both variants, we set B_k to be the canonical basis so that $B_k \subset D_k$ in the first case and $B_k \cap D_k = \emptyset$ in the second.



(a) $\epsilon_f = 10^{-3}$.



(b) $\epsilon_f = 10^{-6}$.

Figure 7.4: Performance of the `dsds` methods (same settings), given a budget of $2000\,n$ evaluations. $\gamma_\alpha = \gamma_\beta$.

The results presented in Figure 7.4 were obtained by giving all four methods the same values for the increase and decrease parameters (i.e. $\gamma_\alpha = \gamma_\beta = \gamma$ and $\theta_\alpha = \theta_\beta = \theta$). The DSDS methods performed relatively poorly compared to the other algorithms: it thus seems that this parameter setting does not favor the first-order aspects. Note however that in terms of robustness, the second variant of Algorithm 7.3 is competitive with the standard direct-search framework, even though it potentially consumes more function evaluations per iteration.

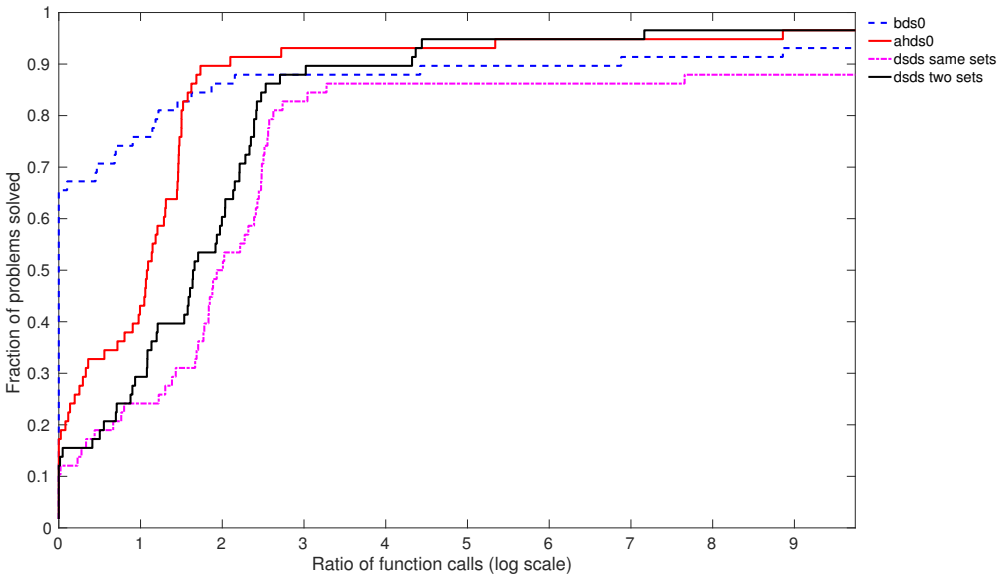
Figure 7.5 presents the results when using different update parameters for the first and second-order polling steps of Algorithm 7.3. We set $\gamma_\beta = \gamma$ but modified the first-order increase parameter so that $\gamma_\alpha > \gamma$; the decrease parameters were identical. With this setting, we were able to achieve comparable and sometimes superior performance than the other algorithms. It is worth noticing that the variant based on different polling sets is the most robust (that is, the best one for large ratios of function evaluations).

Two conclusions can thus be drawn from such experiments. The first one is the interest of fully de-coupling the first and second-order aspects by means of different polling sets: for nonconvex problems, it seems preferable to rely on symmetric polling sets, which confirms our findings from Chapter 6. The second remark one can make is related to the values of the update parameters, that appear to be critical for designing an efficient de-coupled method. Even though no a priori information is available for tuning these parameters, one could consider their dynamical adjustment (which would not compromise the theory as long as the parameters vary in a bounded interval) to avoid excessive decrease in one of the step sizes, which appears to be the case on our first experiment.

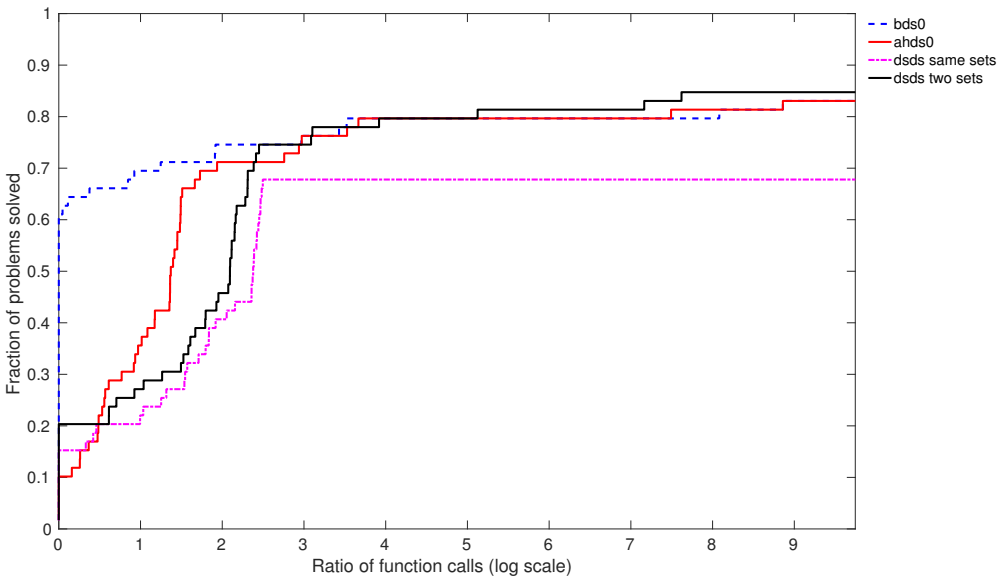
In the probabilistic setting, we have some insights on the choice of such parameters that are not tailored to specificities of the considered problem(s), but rather to the satisfaction of the probabilistic descent property described in Assumption 7.9. Indeed, we can set γ_α and θ_α to yield $p_0 = 0.5$, which then means that D_k can be chosen by taking one direction uniformly generated on the sphere and its opposite.

Figures 7.6 to 7.8 present a comparison between two deterministic methods, `bds0` and `ahds0` introduced in Chapter 6, and two randomized methods. The first one is called `dspd` (direct search based on probabilistic descent) and corresponds to the method described in Chapter 3, with a forcing function $\rho(\alpha) = c\,\alpha^3$ to cope with second-order aspects. The second method is a variant of Algorithm 7.3 based on a random first-order polling step, where the canonical basis used in the second-order polling step: it is called `dsds 1st-rand` in the figures. Both of these methods use polling sets made of two opposite directions, as we just described. One sees that the performance of `dspd` is remarkably good in this setting, thereby confirming the numerical observations of Chapter 3. Nevertheless, it appears that the de-coupled strategy brings more robustness to the method. As the required precision in terms of function values increases, one can observe that `dsds 1st-rand` gets closer in performance to `dspd`, and that it is competitive with `ahds0` in terms of robustness.

As a final remark, note that we did not propose a variant of the `ahds` method based on random directions instead of the initial polling set since such an approach would modify



(a) $\epsilon_f = 10^{-3}$.



(b) $\epsilon_f = 10^{-6}$.

Figure 7.5: Performance of the `dsds` methods (same settings), given a budget of $2000\,n$ evaluations. $\gamma_\alpha > \gamma_\beta$.

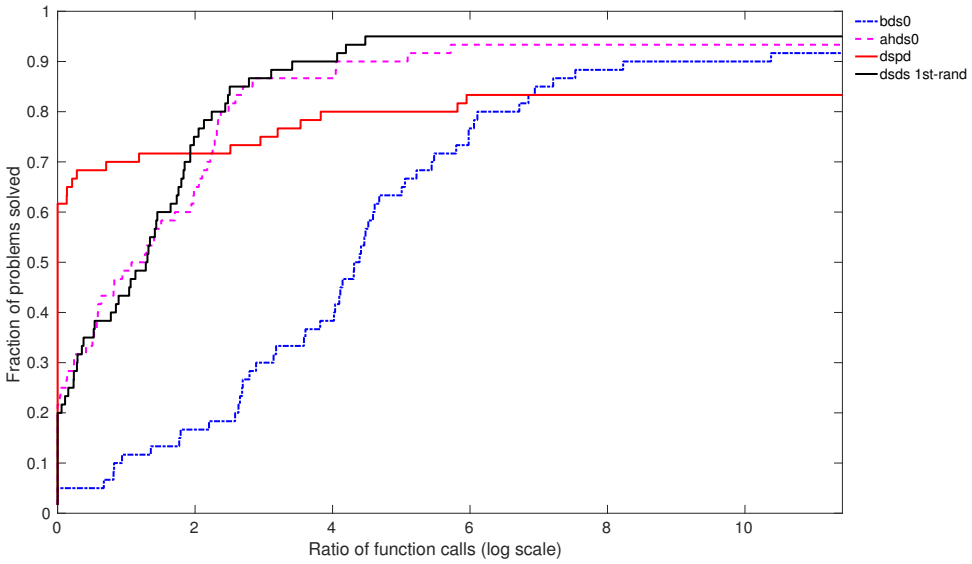


Figure 7.6: Performance of deterministic and probabilistic methods given a budget of $2000\,n$ evaluations, $\epsilon_f = 10^{-3}$.

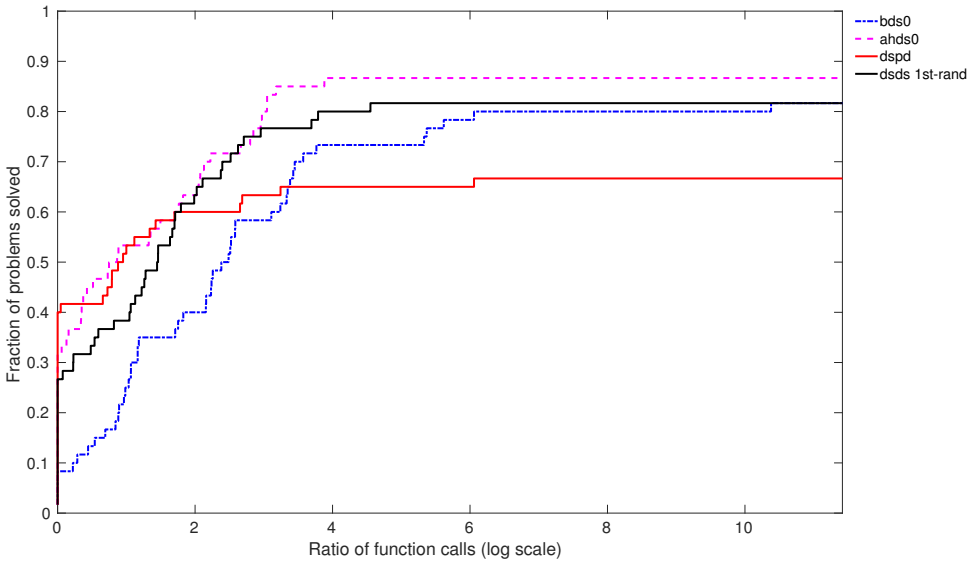


Figure 7.7: Performance of deterministic and probabilistic methods given a budget of $2000\,n$ evaluations, $\epsilon_f = 10^{-6}$.

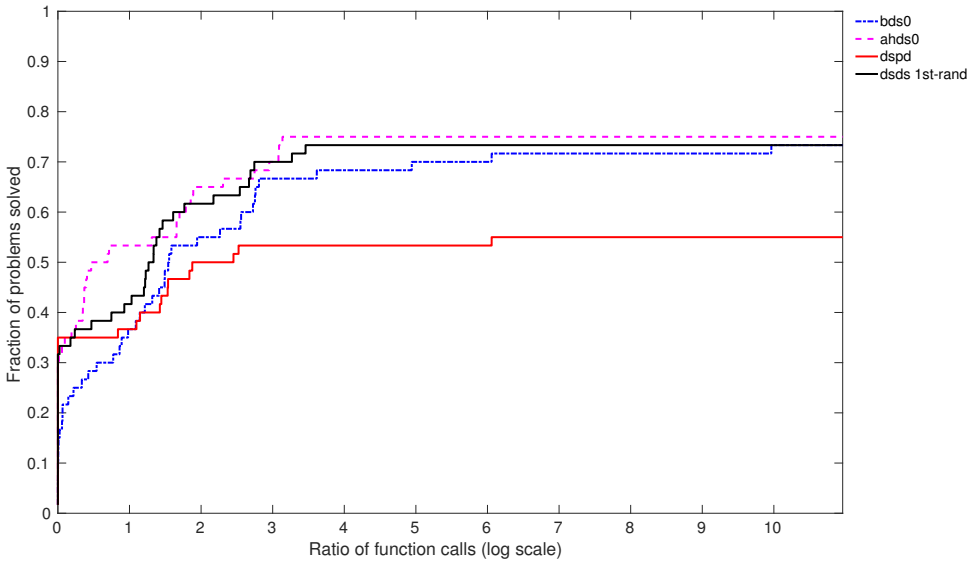


Figure 7.8: Performance of deterministic and probabilistic methods given a budget of $2000\,n$ evaluations, $\epsilon_f = 10^{-9}$.

the algorithmic framework. Indeed, in Algorithm 6.1, we aim at reusing the values we compute in the first part of the polling step by extracting a linear basis from the initial polling set. This is possible whenever this set is a PSS (which we assumed in the deterministic setting of Chapter 6), however using random directions would annihilate this property, thus requiring additional computational effort to be performed for building the approximate Hessian.

7.6 Conclusions for Chapter 7

We have proposed a technique that separates the first and second-order aspects of a second-order globally convergent derivative-free methods, which we have used to define new variants on the trust-region and direct-search frameworks. By giving priority to first-order considerations, we are able to save function evaluations compared to classical implementations, while accounting for the possible scaling variations between the first-order and the second-order criteria. Interestingly, this approach improves the existing worst-case complexity bounds for second-order optimality, even when compared to certain derivative-based algorithms.

The introduction of randomness within the de-coupled strategies was also investigated, and illustrated using the two de-coupled derivative-free algorithms we defined. The resulting probabilistic analysis shows that second-order global convergence can be guaranteed with probability one, and that complexity guarantees are preserved with high

probability.

A series of numerical experiments was finally conducted to assess the interest of the proposed approaches. The deterministic instances of the de-coupled strategies were observed to improve the usual variants, provided careful parametrization was used, so as to deal with each criterion in an appropriate fashion. In particular, the direct-search de-coupled scheme revealed itself very sensitive to such aspects. This issue was partially addressed thanks to probabilistic requirements, yielding a increase in both performance and robustness of the algorithm.

Chapter 8

Probabilistic second-order descent

Second-order aspects have been introduced to improve the performance of optimization algorithms that initially relied on first-order properties. Indeed, they can significantly improve optimization methods, in particular when the function to optimize is nonconvex. However, taking second order into account usually represents a significant increase in the cost of the algorithm, which is why algorithms have also incorporated second-order information without enforcing second-order convergence. The resulting methods may still prove effective, and partial explanation for such a behavior may be obtained through a probabilistic, second-order analysis of these features.

In this chapter, we define probabilistic properties that express the ability of a given direction to capture (negative) curvature information. Section 8.1 reviews the second-order convergence proofs that have been proposed in the derivative-based literature, and identifies the common second-order requirements for the steps that are used. A class of such vectors is described in Section 8.2, and illustrated by specific examples corresponding to known requirements in both derivative-free and derivative-based algorithms. Provided this property can be ensured with sufficiently high probability, theoretical convergence follows: this is established in Section 8.3 using a direct-search framework. In Sections 8.4 and 8.5, practical satisfaction of particular cases through random generation is discussed.

8.1 Second-order convergence in numerical optimization

The motivation of the present chapter originates from the derivative-based literature. Indeed, for these methods, it is not uncommon to require convergence towards a point satisfying second-order necessary optimality conditions. Trust-region algorithms have had a long tradition of such requirements [93, 107], so have line-search methods, often called *curvilinear search* algorithms in this context [62, 94]. More recently, the computation of negative curvature directions has been subject to renewed investigation in the line-search context [11, 12, 63].

Second-order properties have also been considered while solving constrained optimization problems. One example is given by Forsgren and Murray [56] who combined descent, negative curvature and constraint-related directions in order to design a second-order convergent Newton method with line search for linearly constrained problems. Second-order necessary conditions have also been derived for trust-region schemes applied to general nonlinearly constrained problems [20, 45].

A common feature of these algorithms is that they rely on the quality of the directions or steps to ensure a decrease in the function value related to the (second-order) convergence criterion, for a sufficiently small step size (a property usually assessed by means of a related sequence of algorithmic parameters). The use of *descent pairs* is characteristic of such techniques: a pair of directions is generated, with one being of descent type and the other of negative curvature. One may then compare or combine those directions, so as to consider a step that possesses both first and second-order characteristics. In a descent pair, the first direction can thus be qualified of first-order descent, in that it can induce descent based upon the first-order optimality criterion, while the other one can be termed second-order descent. In this chapter, we are particularly interested in the second type of directions, and in the way it can complement the computation of a first-order one.

8.2 A class of second-order probabilistic properties

In this section, we define a general property that characterizes so-called directions of *second-order descent*. Its purpose is to identify vectors that make use of (significant) negative curvature information so as to induce decrease in the function value when taken with a sufficiently small step size. Such a definition encompasses commonly used directions, such as negative curvature ones and approximations of those.

Our goal is to determine if a given direction is able to gather information related to negative curvature. Consider thus a given point $x \in \mathbb{R}^n$, and suppose that $\nabla^2 f(x)$ has at least one negative eigenvalue $\lambda = \lambda_{\min}\left(\nabla^2 f(x)\right) < 0$. Given a step size $\alpha > 0$, we will say that a direction $d \in \mathbb{R}^n$ is a *second-order descent* direction if

$$c_1\, \alpha\, \nabla f(x)^\top\, d + \frac{\alpha^2}{2} d^\top\, \nabla^2 f(x)\, d \;\leq\; c_2\, \frac{\alpha^2}{2}\, \lambda + c_3\, \alpha^3, \tag{8.1}$$

where $c_1 \geq 0,\ c_2 \in (0,1),\ c_3 \geq 0$.

The purpose of the generic formulation of (8.1) is threefold. First, the left-hand side of the equation is related to the second-order Taylor expansion of $f(x+\alpha\, d)$, which is the basic analytical tool in a smooth setting. In addition, the criterion of interest, λ, appears in the right-hand side multiplied by α^2 (the same order as $d^\top \nabla^2 f(x)\, d$). Finally, the last part of the right-hand side can be viewed as an error term, which is one order higher in terms of power of α than the one involving λ. Specific cases of this property will essentially be obtained by acting on c_1 and c_3, as we will study in Sections 8.4 and 8.5. Beforehand, we describe how a probabilistic variant on (8.1) can lead to almost-sure second-order convergence.

8.3 Probabilistic analysis of a direct-search scheme based on second-order descent

In order to derive a second-order probabilistic analysis, we again consider Algorithm 3.1. However, we will now require a different property to be satisfied (with a given probability) by the polling sets. We begin by defining its deterministic counterpart.

Definition 8.1 *Consider a realization of Algorithm 3.1 applied to minimize a function f satisfying Assumption 6.1. Let $\kappa_g \in (0,1), \kappa_H \in (0,1], c_H \geq 0$; we say that at the k-th iteration, the set D_k is* (κ_g, κ_H)-second-order descent *if either $\|g_k\| \geq -\lambda_k$ and*

$$\mathrm{cm}(D_k, -g_k) \;\geq\; \kappa_g, \tag{8.2}$$

or $-\lambda_k > \|g_k\|$ and

$$\exists d \in D_k,\; \alpha_k\, g_k^\top\, d + \frac{\alpha_k^2}{2} d^\top\, \nabla^2 f(x_k)\, d \;\leq\; \kappa_H \frac{\alpha_k^2}{2} \lambda_k + \frac{c_H\, \nu_H}{6} \alpha_k^3, \tag{8.3}$$

with $c_H \geq 0$ does not depend on k and ν_H is a Lipschitz constant for $\nabla^2 f$.

Note that (8.3) is a particular case of (8.1) with $c_1 = 1$. The general case with $c_1 \in [0,1]$ could also be considered, typically by exploiting symmetric directions to ensure that $g_k^\top\, d \leq 0$ for one selected vector.

The second-order descent property expresses the fact that we look for decrease based on the most promising criterion. If the gradient norm is higher than the opposite minimum eigenvalue, then it is likely that we will obtain a greater decrease using a direction of first-order descent, that is making an acute angle with the negative gradient; in that case, what we are requiring for the set D_k is the κ_g-descent property, in the sense of Definition 3.1. When $-\lambda_k > \|g_k\|$, however, we would like to take a direction possibly yielding a decrease associated with the negative curvature, and this is the purpose of (8.3). The corresponding direction is asymptotically an approximate curvature one, as the term in α_k^3 quantifies the error committed while estimating such a direction.

Definition 8.2 *Let $p \in (0,1], \kappa_g \in (0,1), \kappa_H \in (0,1], c_H \geq 0$.*
Given the sequence of iterates and polling sets produced by Algorithm 3.1, let E_k be the

event $\{\mathfrak{D}_k$ *is* (κ_g, κ_H)*-second-order descent*$\}$.
The sequence $\{\mathfrak{D}_k\}_k$ *is called a p-probabilistically* (κ_g, κ_H)*-second-order descent sequence, or a* (p, κ_g, κ_H)-second-order descent sequence *in short, if*

$$\mathbb{P}\left(E_0\right) \;\geq\; p \quad \textit{and} \quad \forall\, k \geq 1,\; \mathbb{P}\left(E_k \mid \sigma\left(\mathfrak{D}_0, \ldots, \mathfrak{D}_{k-1}\right)\right) \;\geq\; p. \tag{8.4}$$

Having identified a probabilistic property, we can proceed as in Section 3.3 to prove that if it is satisfied with a sufficiently high probability at every iteration, then Algorithm 3.1 will converge almost surely to a second-order stationary point.

In addition to the previous definitions, we define two functions φ_g and φ_H that will intervene in our analysis the same way as the function φ in Chapter 3.

$$\forall\, t > 0,\; \left\{ \begin{array}{lcl} \varphi_g(t) & = & \inf\left\{\alpha > 0 \;\middle|\; \frac{\rho(\alpha)}{\alpha} + \frac{\nu_g}{2}\,\alpha \geq t\right\} \\ \\ \varphi_H(t) & = & \inf\left\{\alpha > 0 \;\middle|\; \frac{2\,\rho(\alpha)}{\alpha^2} + \frac{\nu_H(1+c_H)}{3}\,\alpha \geq t\right\}. \end{array} \right. \tag{8.5}$$

If ρ satisfies Assumption 6.2 (e.g. $\rho(\alpha) = \alpha^3$), we are guaranteed that these functions are well-defined for every $t \in (0, +\infty)$ and non-decreasing on this interval.

Using φ_g and φ_H, we obtain a result concerning unsuccessful iterations as follows.

Lemma 8.1 *Consider the k-th iteration of a realization of Algorithm 3.1 applied to a function satisfying the smoothness requirements of Assumption 6.1, and let Assumption 6.2 hold. If* D_k *is* (κ_g, κ_H)*-second-order descent and*

$$\alpha_k \;<\; \min\left\{\varphi_g(\kappa_g\,\tau_k), \varphi_H(\kappa_H\,\tau_k)\right\}, \tag{8.6}$$

where $\tau_k = \max\left\{\|g_k\|, -\lambda_k\right\}$*, the k-th iteration is successful.*

Proof. Suppose first that $\|g_k\| \geq -\lambda_k$. Since D_k is (κ_g, κ_H)-second-order descent, it means that $\mathrm{cm}(D_k, -g_k) > \kappa_g$. Thus, there exists $d_1 \in D_k$ such that one can derive :

$$f(x_k + \alpha_k\, d_1) - f(x_k) \;\leq\; -\kappa_g\,\alpha_k\,\|g_k\| + \frac{\nu_g}{2}\alpha_k^2, \tag{8.7}$$

and given that α_k satisfies (8.6), we have that $f(x_k + \alpha_k\, d_1) - f(x_k) < -\rho(\alpha_k)$, so k is successful.

Suppose now that $-\lambda_k > \|g_k\|$ (so $\lambda_k < 0$); then there exists $d_2 \in D_k$ satisfying (8.3), which means that one has:

$$\begin{array}{rcl} f(x_k + \alpha_k\, d_2) - f(x_k) & \leq & \alpha_k\, g_k^\top\, d_2 + \frac{\alpha_k}{2} d_2^\top\, \nabla^2 f(x_k)\, d_2 + \frac{\nu_H\,\alpha_k^3}{6} \\ & \leq & \kappa_H\,\frac{\alpha_k^2}{2}\lambda_k + \frac{\nu_H(1+c_H)\,\alpha_k^3}{6}. \end{array}$$

Again using (8.6), we have that $f(x_k + \alpha_k\, d_2) - f(x_k) < -\rho(\alpha_k)$, thus k is a successful iteration. □

We can now reproduce the reasoning of Chapter 3 in order to establish convergence, by means of submartingale arguments. The main result is stated in the following theorem.

Theorem 8.1 *Consider the application of Algorithm 3.1 under Assumptions 2.2, 6.1 and 6.2. If $\{\mathfrak{D}_k\}$ is (p, κ_g, κ_H)-second-order descent with $p \geq p_0$, with $p_0 = \ln \theta / \ln(\gamma^{-1}\,\theta)$, then*

$$\mathbb{P}\left(\liminf_{k\to\infty} \max\{\|\nabla f(X_k)\|, -\Lambda_k\} = 0\right) \;=\; 1. \tag{8.8}$$

We are now interested in the complexity results that arise from the use of probabilistic second-order descent properties. Our goal is to bound (in probability) the number of iterations and function evaluations needed to achieve

$$\left\|\tilde{G}_k\right\| < \epsilon_g \text{ and } \tilde{\Lambda}_k > -\epsilon_H, \tag{8.9}$$

where $\tilde{G}_k$ and $\tilde{\Lambda}_k$ denote the gradient with minimal (positive) norm and the maximum of the (negative) minimum Hessian eigenvalues along the $k+1$ first iterations, respectively. Their realizations will be denoted by $\tilde{g}_k$ and $\tilde{\lambda}_k$.

For the rest of this section, we will denote by K_ϵ the first index such that (8.9) holds for $K_\epsilon + 1$, indicating its realizations by k_ϵ.

Lemma 8.2 *Let the assumptions of Theorem 8.1 hold. Given a realization of our algorithm and a positive integer $k \leq k_\epsilon$,*

$$\sum_{l=0}^{k-1} z_l \;\leq\; \frac{\beta}{\rho\left(\min\left\{\gamma^{-1}\,\alpha_0, \varphi_g(\kappa_g\,\epsilon_g), \varphi_H(\kappa_H\,\epsilon_H)\right\}\right)} + p_0\,k, \tag{8.10}$$

where β is a bound on $\sum_{i=0}^{\infty} \rho(\alpha_i)$, that can be set as in Lemma 3.7.

Lemma 8.2 is the second-order equivalent to Lemma 3.8. Its result is the key argument to establish the complexity rate for a probabilistic second-order descent sequence. Note that if deterministic second-order descent can be ensured, we recover a complexity result similar to that of Algorithm 6.1 (AHDS).

Theorem 8.2 *Under the assumptions of Theorem 8.1, suppose that the sequence $\{\mathfrak{D}_k\}$ is (p, κ_g, κ_H)-second-order descent with $p > p_0$ and let $(\epsilon_g, \epsilon_H) \in (0,1)$ chosen such that*

$$\epsilon_g < \kappa_g^{-1}\left[\frac{\rho(\gamma^{-1}\,\alpha_0)}{\gamma^{-1}\,\alpha_0} + \frac{\nu_g\,\gamma^{-1}\,\alpha_0}{2}\right] \tag{8.11}$$

and

$$\epsilon_H < \kappa_H^{-1}\left[\frac{2\,\rho(\gamma^{-1}\,\alpha_0)}{\gamma^{-2}\,\alpha_0^2} + \frac{\nu_H(1+c_H)\,\gamma^{-1}\,\alpha_0}{3}\right] \tag{8.12}$$

hold. Then, for each $\delta > 0$, the probability

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \frac{(1+\delta)\beta}{(p-p_0)\min\left\{\rho[\varphi_g(\kappa_g\,\epsilon_g)], \rho[\varphi_H(\kappa_H\epsilon_H)]\right\}} \right\rceil\right)$$

is higher than or equal to

$$1 - \exp\left[-\frac{\beta(p-p_0)\delta^2}{2\,p(1+\delta)\min\left\{\rho[\varphi_g(\kappa_g\,\epsilon_g)],\rho[\varphi_H(\epsilon_H)]\right\}}\right].$$

We illustrate the results of Theorem 8.2 for a specific choice of forcing function.

Corollary 8.1 *Let the assumptions of Theorem 8.1 hold. If the sequence* $\{\mathfrak{D}_k\}$ *is* (p,κ_g,κ_H)*-second-order descent with* $p > p_0$*,* $\rho(\alpha) = c_\rho\,\alpha^3/6$*,*

$$\epsilon_g \;\leq\; \frac{1}{\kappa_g}\,\min\left\{\frac{3\,c_\rho}{\nu_g^2}, \frac{c_\rho}{6}\gamma^{-2}\,\alpha_0^2 + \frac{\nu_g}{2}\gamma^{-1}\,\alpha_0\right\} \tag{8.13}$$

and

$$\epsilon_H \;\leq\; \frac{(c_\rho + \nu_H(1+c_H))}{3\,\kappa_H}\gamma^{-1}\,\alpha_0. \tag{8.14}$$

then one has

$$\mathbb{P}\left(K_\epsilon \leq \left\lceil \tfrac{\mathcal{C}_1}{p-p_0}\max\left(\kappa_g^{-3}\,\epsilon_g^{-3}, \kappa_H^{-3}\epsilon_H^{-3}\right)\right\rceil\right) \geq 1 - \exp\left[-\tfrac{(p-p_0)\mathcal{C}_2}{p}\max\left(\kappa_g^{-\frac{3}{2}}\,\epsilon_g^{-\frac{3}{2}}, \kappa_H^{-3}\epsilon_H^{-3}\right)\right] \tag{8.15}$$

with $\mathcal{C}_1 = \frac{12\,\beta}{c_\rho}\,\max\left(\nu_g^3, \frac{(c_\rho+\nu_H(1+c_H))^3}{27}\right)$ *and* $\mathcal{C}_2 = \frac{3\,\beta}{2\,c_\rho}\max\left((c_\rho/6)^{\frac{3}{2}}, \frac{(c_\rho+\nu_H(1+c_H))^3}{27}\right)$.

Proof. From the definition of φ_H, it is clear that $\varphi_H(t) = 3\,t/(c_\rho + \nu_H(1+c_H))$. As for the expression of $\varphi_g(t)$, a second-order polynomial calculation yields for all $t > 0$:

$$\varphi_g(t) \;=\; \frac{3}{c_\rho}\left[-\frac{\nu_g}{2} + \sqrt{\frac{\nu_g^2}{4} + \frac{2}{3}c_\rho\,t}\right]. \tag{8.16}$$

For every $0 < t < \frac{3\,c_\rho}{\nu_g^2}$, one obtains the following bounds on $\varphi_g(t)$:

$$\left\{\begin{array}{lcl} \varphi_g(t) & \leq & \phi_{up}(t) = \frac{3}{c_\rho}\sqrt{\frac{2}{3}c_\rho\,t} \\ \varphi_g(t) & \geq & \phi_{low}(t) = \frac{t}{\nu_g}. \end{array}\right. \tag{8.17}$$

Let p_{K_ϵ} denote the probability

$$\mathbb{P}\left(K_\epsilon \leq \frac{2\,\beta}{p-p_0}\frac{1}{\min\left\{\rho\left(\varphi_g(\kappa_g\,\epsilon_g)\right), \rho\left(\varphi_H(\kappa_H\,\epsilon_H)\right)\right\}}\right).$$

Then, since ρ is non-decreasing, this probability is bounded above as follows:

$$\begin{array}{rcl} p_{K_\epsilon} & \leq & \mathbb{P}\left(K_\epsilon \leq \dfrac{2\,\beta}{p-p_0}\dfrac{1}{\min\left\{\rho\left(\phi_{low}(\kappa_g\,\epsilon_g)\right), \rho\left(\varphi_H(\kappa_H\,\epsilon_H)\right)\right\}}\right) \\ & \leq & \mathbb{P}\left(K_\epsilon \leq \left\lceil \dfrac{\mathcal{C}_1}{p-p_0}\max\left(\kappa_g^{-3}\,\epsilon_g^{-3}, \kappa_H^{-3}\epsilon_H^{-3}\right)\right\rceil\right). \end{array}$$

Since we are in the assumptions of Theorem 8.2 with $\delta = 1$, we also have

$$\begin{aligned} p_{K_\epsilon} &\geq 1 - \exp\left[-\frac{\beta(p-p_0)}{4\,p\min\left\{\rho[\varphi_g(\kappa_g\,\epsilon_g)],\rho[\varphi_H(\epsilon_H)]\right\}}\right] \\ &\geq 1 - \exp\left[-\frac{\beta(p-p_0)}{4\,p\min\left\{\rho[\phi_{up}(\kappa_g\,\epsilon_g)],\rho[\varphi_H(\epsilon_H)]\right\}}\right] \\ &\geq 1 - \exp\left[-\frac{(p-p_0)\mathcal{C}_2}{p}\max\left(\kappa_g^{-\frac{3}{2}}\,\epsilon_g^{-\frac{3}{2}},\kappa_H^{-3}\epsilon_H^{-3}\right)\right]. \end{aligned}$$

Combining the two relations leads to (8.15). □

We thus obtain a complexity in $\mathcal{O}\left(\max\left\{\kappa_g^{-3}\,\epsilon_g^{-3},\kappa_H^{-3}\epsilon_H^{-3}\right\}\right)$ in terms of powers of the tolerances with overwhelming probability. Note that unlike in the first-order result (3.47) of Chapter 3, the powers of the tolerances are not the same on both sides of the equation (8.15).

8.4 Full second-order properties

We now go back to (8.1) and look at instances of the property for which $c_1 = 0$. Those cases can be qualified as *full second-order* since they only involve the second-order derivative of the objective function.

8.4.1 The case of negative curvature directions

In the specific case of $c_1 = c_3 = 0$, (8.1) reduces to

$$d^\top \nabla^2 f(x)\, d \;\leq\; c_2\,\lambda \;<\; 0, \tag{8.18}$$

which means that d is a direction of negative curvature. We recall that when one aims to exploit negative curvature, one typically considers directions satisfying

$$d^\top \nabla f(x) \leq 0 \quad \text{and} \quad d^\top \nabla^2 f(x)\, d \leq c_2\,\lambda. \tag{8.19}$$

It is known that the computation of such a vector can be an expensive process in a large-scale setting, as it either requires to solve of an eigenvalue problem or to use linear algebra methods [18, Section 4.3 and Chapter 5]. In a derivative-free context, it is almost impossible to generate directions that satisfy (8.19). In fact, in a deterministic setting, one can always construct matrices for which the cone of (sufficient) negative curvature directions does not intersect a given polling set. Ensuring that this intersection is not empty in probability may be too expensive as it would necessitate to use an increasingly large number of directions. We provide insights on this remark in the next paragraph, for the specific case of standard normal and uniform distributions.

The Rayleigh quotient and its distribution Given the results presented in Chapter 3, it is a natural question to ask whether the uniform distribution exhibits desirable second-order properties. The probability distribution of the Rayleigh quotient in the case of a uniformly distributed direction has been the subject of several works [18, 71, 72]. We recall below the formula established by Boman [18].

Theorem 8.3 *Let $t \in [\lambda_1, \lambda_n]$, where $\lambda_1 \leq \cdots \leq \lambda_n$ are the eigenvalues of the matrix A; then, if d is a random vector uniformly distributed in the unit sphere or follows a standard normal distribution in $\mathbb{R}^n$, one has*

$$\mathbb{P}\left(\frac{d^\top A\, d}{\|d\|^2} \leq t\right) = \frac{2}{\pi}\int_0^\infty \frac{\cos\left(\frac{1}{2}\sum_{\lambda_j > t}\tan^{-1}((\lambda_j - t)u\right)\sin\left(\frac{1}{2}\sum_{\lambda_j < t}\tan^{-1}((t-\lambda_j)u)\right)}{u\prod_{j=1}^n (1+(\lambda_j - t)^2\, u^2)^{\frac{1}{4}}}\, du. \tag{8.20}$$

Using this distribution, we can design an example of pathological behavior for the Rayleigh quotient. For simplicity, we consider $c_2 = \frac{1}{2}$, but similar results hold with any $c_2 \in (0, 1)$.

Let $\{H_k\}$ be a sequence of 2-by-2 matrices such that for every k, the eigenvalues of H_k are $\lambda_{max} = 1$ and $\lambda_k = -\frac{1}{k+1}$. We then have the following result.

Lemma 8.3 *Let $\{H_k\}_k$ be a sequence of 2-by-2 matrices such that for all k, the eigenvalues of H_k are $\lambda_{max} = 1$ and $\lambda_k = -\frac{1}{k+1}$. Let d be a vector uniformly distributed in the unit sphere; then*

$$\lim_{k\to\infty} \mathbb{P}\left(d^\top H_k\, d < \frac{\lambda_k}{2}\right) = 0. \tag{8.21}$$

Proof. It directly follows from Theorem 8.3 that for every k

$$\mathbb{P}\left(d^\top H_k\, d < \frac{\lambda_k}{2}\right) \;=\; \frac{2}{\pi}\int_0^\infty \frac{\cos\left(\frac{1}{2}\tan^{-1}(\frac{2k+1}{2k+2}u)\right)\sin\left(\frac{1}{2}\tan^{-1}(\frac{1}{2k+2}u)\right)}{u\left(1+\frac{(2k+1)^2}{(2k+2)^2}u^2\right)^{\frac{1}{4}}\left(1+\frac{1}{(2k+2)^2}u^2\right)^{\frac{1}{4}}} du.$$

When $k \to \infty$, the term in the sinus goes to zero while the denominator does not, hence the result. □

We will make use of $p_k \equiv \mathbb{P}\left(d^\top H_k\, d < \lambda_k/2\right)$ to show that when each iteration uses a finite number of uniform directions, the probability of satisfying (8.19) goes to zero.

Theorem 8.4 *Let $\{\mathfrak{D}_k\}_k$ be a sequence of randomly, independently generated sets of m vectors uniformly distributed in the unit sphere, and let $\{H_k\}$ be the same sequence of matrices as in Lemma 8.3. Then, we necessarily have*

$$\lim_{k\to\infty} \mathbb{P}\left(\min_{d\in\mathfrak{D}_k} d^\top H_k\, d < \frac{\lambda_k}{2}\right) \;=\; 0. \tag{8.22}$$

Proof. For every k, we have that

$$\mathbb{P}\left(\min_{d\in\mathfrak{D}_k} d^\top H_k\, d < \frac{\lambda_k}{2}\right) \;=\; 1 - (1 - p_k)^m$$

where p_k is the probability described in Lemma 8.3. Thanks to Lemma 8.3, we know that the sequence $\{p_k\}$ goes to zero independently of $\{\mathfrak{D}_k\}$. As a result, (8.22) holds. □

It is our belief that one can define a pathological objective for which the optimization process would require an increasing number of directions for the sequence to be probabilistically second-order descent (here meaning that one direction is of negative curvature). However, such an example would be quite more difficult to construct than the deterministic counter-examples built by Cartis, Gould and Toint while dealing with complexity issues [22, 27], as one would have to guarantee that the path taken by the iterates would satisfy desirable properties for a set of realizations of the algorithm with a non-zero measure.

8.4.2 A particular case based on a randomized Hessian approximation

Suppose now that $c_1 = 0$, but $c_3 > 0$; in that case (8.1) becomes

$$d^\top \nabla^2 f(x)\, d \;\leq\; c_2\, \lambda + 2\, c_3\, \alpha. \tag{8.23}$$

Provided c_3 is appropriately chosen, one thus obtains that d is required to be an *approximate negative curvature direction*, in the sense of Proposition 6.1.

In derivative-free optimization, one can find several examples of second-order convergent algorithms that aim to compute such directions, typically basing themselves on Hessian approximations [38, 57, 66]. The quality of the Hessian approximation is then a crucial argument for proving convergence and represents the main concern of these methods. Ensuring this quality is often expensive in terms of function evaluations. Bandeira et al [14] considered probabilistically *fully quadratic* models for which the Hessian approximation at the k-iteration $\mathcal{H}_{m_k}$ was supposed to satisfy

$$\|\mathcal{H}_{m_k} - \mathcal{H}_k\| \leq \mathcal{O}(\Delta_k), \tag{8.24}$$

Δ_k being the random trust-region radius, with probability bigger than 0.5. As a matter of fact, this is the same approximation order than the one in terms of the step size required by Abramson et al [6] for an estimation of the Hessian. The same order also appeared in the Hessian approximation built in Chapter 6.

Note that ensuring (8.23) is more suitable in frameworks such as those of Algorithms 6.1 or 7.3. However, we point out that the basic framework of Algorithm 2.1 could compute polling sets based on an Hessian approximation (even if the related expense would be prohibitive).

Consider the convergence theory detailed in Chapter 6 for Algorithm 6.1. A key feature of this analysis is the ability of the method to consider an approximate direction of negative curvature, as shown by Proposition 6.1. This property can also be considered in a probabilistic context, thanks to the following result from random matrix theory.

Lemma 8.4 *Let $\mathcal{B}$ be an n-by-n matrix with independent entries generated following a standard normal distribution. Then, for any $\sigma > 0$,*

$$\mathbb{P}\left(\sigma_{\min}(\mathcal{B})^2 \geq \sigma\right) \;\geq\; 1 - \frac{2^{-1/2}\,\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)\,\Gamma(1)}\,\mathrm{e}^{-\frac{\sqrt{\sigma}}{2}}\,\sigma^{-\frac{1}{4}}. \tag{8.25}$$

Proof. From the distribution of $\mathcal{B}$, we know that the matrix $W = \mathcal{B}^\top \mathcal{B}$ is a real *Wishart matrix*. As a result, we aim at bounding

$$\mathbb{P}\left(\lambda_{\min}(W) \geq \sigma^{\frac{1}{2}}\right).$$

Applying [31, Lemma 3.3] then yields the result. □

In the two previous chapters, we have seen methods relying on a Hessian approximation computed from a linear basis B_k. Provided $\sigma(B_k)^2 \geq \sigma$ for some $\sigma \in (0, 1]$, the quality of the approximation was ensured. Thanks to Lemma 8.4, we now have a practical way of satisfying this property in probability, without increasing the number of considered vectors.

As a result, one could define a probabilistic property for the B_k matrices to be used in the framework of Algorithm 6.1 (or 7.3).

8.5 Practical satisfaction of mixed first and second-order properties

We now investigate the applicability of (8.1) in a general setting, considering a point x at which both $\|\nabla f(x)\| \neq 0$ and $\lambda_{\min}\left(\nabla^2 f(x)\right) < 0$ hold.

Our goal is to provide experimental justification that a simple randomized approach (e.g. by generating random directions uniformly on the unit sphere) is already capable of producing directions satisfying the second-order-descent property.

8.5.1 Experiments on toy quadratics

We first present numerical insights corresponding to (8.3) (i.e., when $c_1 = 1,\ c_3 > 0$ and $\|\nabla f(x)\| \neq 0$). To do so, we performed a series of tests on quadratic functions.

We conducted two series of experiments. The first one consisted in considering quadratic forms and randomly generating couples of opposite directions from a standard normal distribution or a uniform one. The quadratic function is of the form

$$Q(d) \ = \ \alpha\, g^\top d + \frac{\alpha^2}{2} d^\top H_\lambda\, d, \tag{8.26}$$

where $g \in \mathbb{R}^n$ and $H_\lambda = \mathrm{diag}(1, \ldots, 1, \lambda)$ with $\lambda < 0$. Note that the spectrum of H_λ is defined so as to represent a pathological case, in which only one eigenvalue is negative, and this value is arbitrarily small in modulus compared to the remaining eigenvalues (normalized for simplicity).

Our objective is to determine

$$\mathbb{P}\left(Q(d) \leq \kappa\, \frac{\alpha^2}{2}\lambda + c_3\, \alpha^3\right). \tag{8.27}$$

Whenever d follows a standard normal or a uniform distribution in $\mathbb{R}^n$, it is possible to reformulate the quadratic function in such a way that it become a sum of independent,

non-centered chi-square and normal/uniform variables. In that setting, one can obtain an explicit expression for (8.27). However, the resulting formula appears practically intractable since it requires the knowledge of the full spectrum (and in particular, of the orders of each eigenvalue) of H_λ to determine the exact distribution [92].

Still, one can try to estimate (8.27) by looking at a large number of directions (in our experiments, we chose 5000 pairs of symmetric directions to be this number). Typical behavior occurred whenever α was small compared to $\frac{\|g\|}{\text{cond}(H_\lambda)}$, with $\text{cond}(H_\lambda)$ being the condition number of the matrix H_λ (in our setting, it thus was $-1/\lambda$). The directions exhibited a fifty-per-cent chance of satisfying (8.27) with $\kappa = \mathcal{O}\left(\frac{1}{n}\right)$. This is illustrated on Figures 8.1 and 8.2, that consider two settings, namely g belonging or being orthogonal to the subspace corresponding to the negative eigenvalue λ.

Interestingly enough, such results hold whether we consider c_3 to be equal to zero or not. This is encouraging in the perspective of satisfying an assumption that would extend the one regarding the cosine measure. Indeed, the κ-descent property of a given set D regarding a (gradient) vector g implies that there exists a direction $d \in D$ such that $\text{cm}(D, -g) = -\frac{d^\top g}{\|d\|\|g\|} \geq \kappa$, which means that

$$\alpha\, d^\top g \;\leq\; -\alpha\, \kappa\, \|d\|\|g\|,$$

for any $\alpha > 0$. We verified on our experiments that by generating symmetric directions, either following a standard normal or uniform distribution, one almost certainly satisfies

$$\alpha\, d^\top g + \frac{\alpha^2}{2} d^\top H\, d \;\leq\; \frac{\alpha^2}{2}\, \kappa\, \lambda$$

when α is chosen sufficiently small.

We also point out that the behavior seems rather independent to the problem dimension. It thus seems that in these settings, one could use less than $\mathcal{O}(n^2)$ directions (which is the order of vectors that have to be considered in the deterministic setting) and still satisfy the second-order descent property with a significant probability. Such a result would match what was obtained regarding first-order descent in Chapter 3.

8.5.2 Practical satisfaction within a direct-search run

We consider here the application of Algorithm 3.1 to the set of 60 nonconvex problems used in Chapter 6. Our goal is to check whether second-order aspects are captured by the random directions that are used. Since derivatives are available through the `CUTEst` interface, we simply verify for each direction computed at iteration k if it satisfies

$$\alpha_k\, d^\top \nabla f(x_k) + \frac{\alpha_k^2}{2} d^\top \nabla^2 f(x_k)\, d \;\leq\; \frac{\alpha_k^2}{2}\, \kappa\, \lambda_{\min}\left(\nabla^2 f(x_k)\right),$$

where we chose $\kappa = 1/n$ so as to match the dependence in n one may encounter while dealing with quadratic polynomial models [38]. The results obtained by using a polling set $\mathfrak{D}_k = [\mathfrak{d} \;\; -\mathfrak{d}]$, with $\mathfrak{d}$ being uniformly distributed over the unit sphere, show that

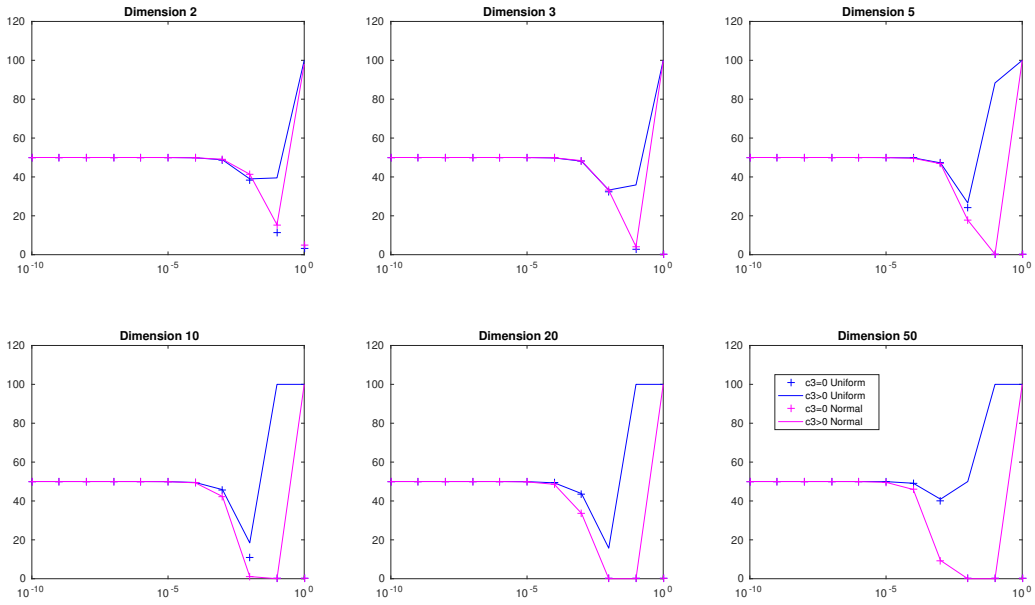
around fifty-per-cent of the directions satisfy the property over all iterations. This number is remarkably higher if one only considers successful directions (that satisfy a sufficient decrease and thus yield a successful iteration), over ninety-per-cent. The percentages are slightly better for the standard normal distribution, but uniform distribution also gives surprisingly good results in those cases. Although we stress out that the percentage values may be influenced by the parametrization of Algorithm 3.1 and by small scaling differences between the two criteria on the tested problems, these results suggest that both distributions are able to generate sequences of probabilistically second-order descent sets.

8.6 Conclusion for Chapter 8

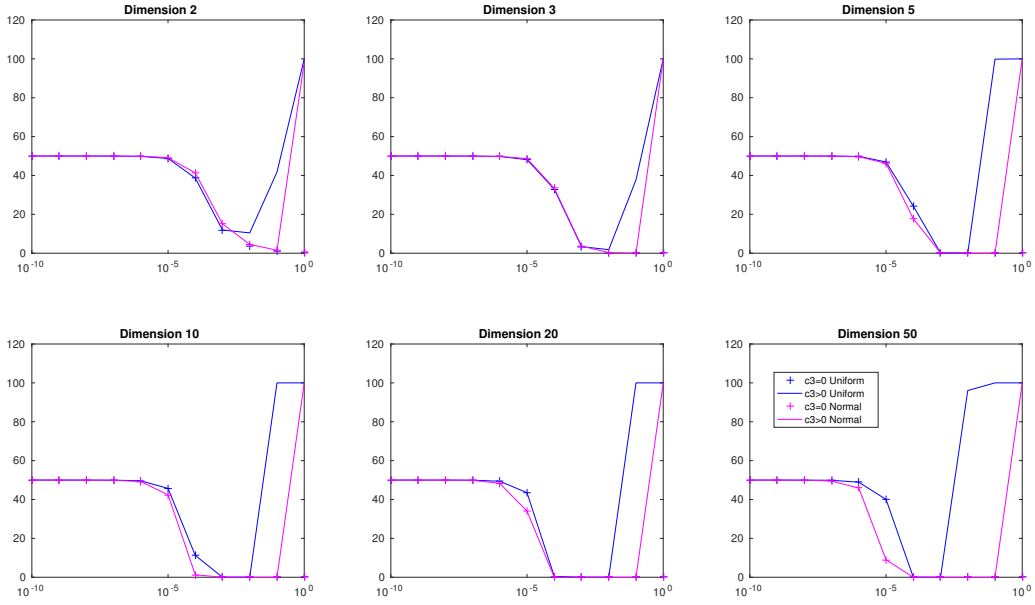
We proposed a generic property that involves the second-order optimality criterion. The general formulation uses the first-order term in the Taylor expansion of the function as well as an error term in order to compensate a possibly too high second-order directional derivative. Using this class of vectors, one can derive second-order convergence and complexity results (in a probabilistic sense) for the direct-search scheme we proposed in Chapter 3. To this end, the polling set sequence is required to satisfy a property that associates the polling requirements to either the first or the second-order criterion, depending on which one that is the most likely to provide decrease.

Although the full characterization of directions satisfying the proposed assumption remains to be done, it has been shown that such a set includes negative curvature directions, as well as approximate eigenvectors of the Hessian matrix. By means of random linear algebra techniques, we can prove that the former (negative curvature property) is rather difficult to achieve by a pure random approach, while the latter (approximate Hessian eigenvector) can be ensured by a specific distribution. The resulting expense is however comparable to a method using an approximate Hessian: as a result, the introduction of randomness does not appear to save function evaluations.

Other specific instances appear surprisingly easy to achieve in practice, as simple probability distributions are observed to produce directions that satisfy the property with a significant average probability. Interestingly, such second-order vectors also seem more likely to produce sufficient decrease, which encourages further their practical use. Besides, since numerical experiments using two directions were quite promising, it is reasonable to believe that certain settings can allow for the use of much less directions than in the case of deterministic second-order globally convergent algorithms.

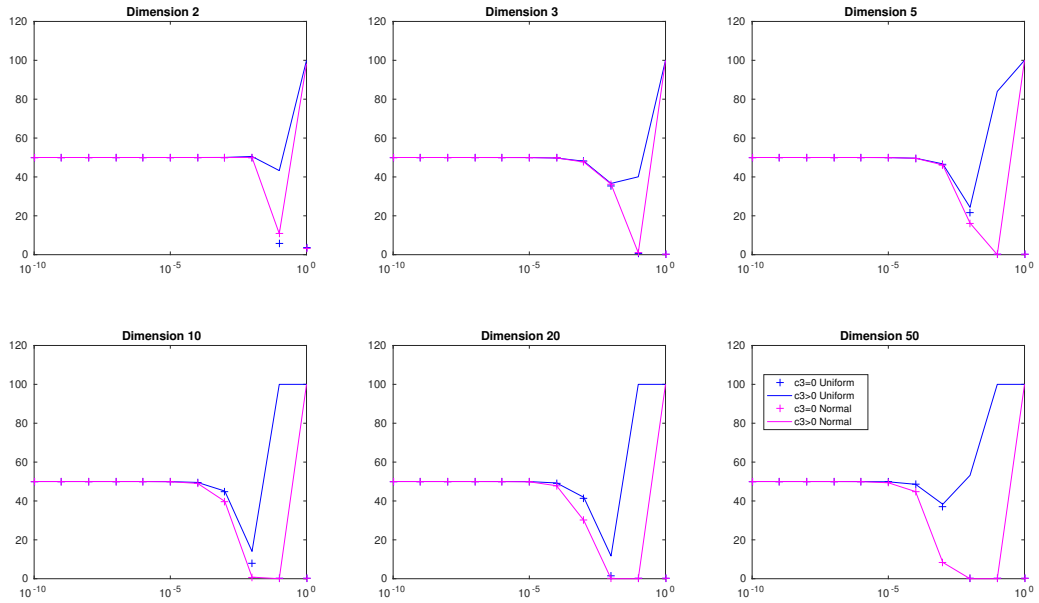


(a) $\varepsilon = 10^{-2}$.

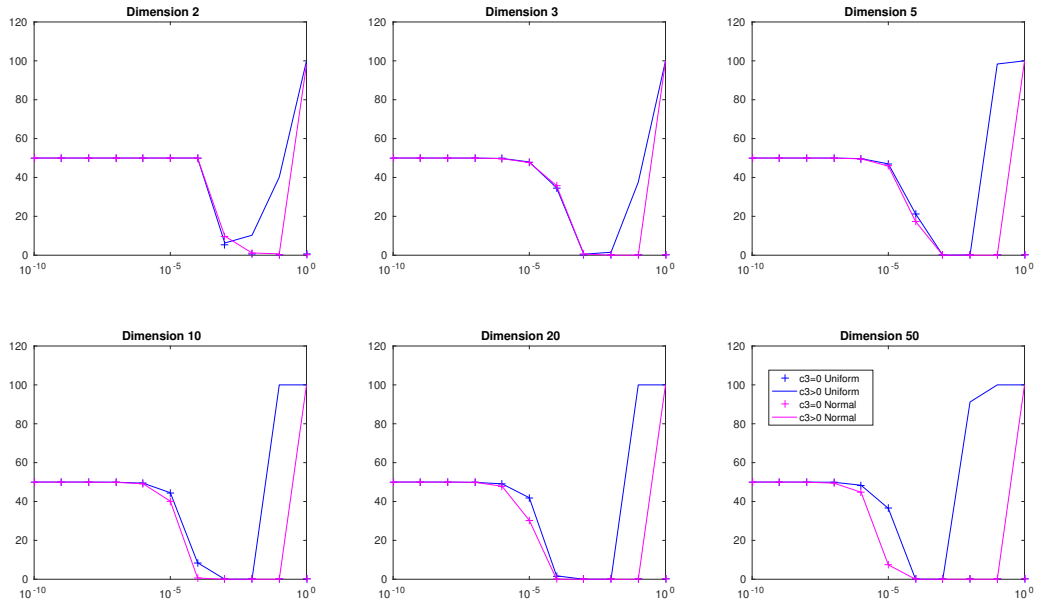


(b) $\epsilon = 10^{-4}$.

Figure 8.1: Percentage of directions satisfying the desired assumptions, for α between 1 and 10^{-10}. Here $\|g\| = \varepsilon^2, \lambda = -\varepsilon$ and $H\,g = \lambda\,g$.



(a) $\varepsilon = 10^{-2}$.



(b) $\epsilon = 10^{-4}$.

Figure 8.2: Percentage of directions satisfying the desired assumptions, for α between 1 and 10^{-10}. Here $\|g\| = \varepsilon^2, \lambda = -\varepsilon$ and g is orthogonal to E_λ.

Chapter 9

Conclusion

Conclusions of the thesis

The first topic of interest for this thesis was the possible benefits in introducing random elements in otherwise deterministic derivative-free algorithmic frameworks. To this end, a probabilistic analysis of a direct-search algorithm based on probabilistically descent directions was derived, endowing such a scheme with convergence guarantees. As the resulting method was observed to outperform its deterministic counterpart, thanks to its potential cheaper cost, it appeared that random aspects could improve the numerical behavior without jeopardizing the convergence of the algorithm. The probabilistic techniques used to design the framework were then adapted to handle simple constraints on the variables. The resulting schemes also yielded significant improvement over a popular deterministic solver. This confirmed the interest of probabilistic properties, both in terms of practical cost and numerical efficiency. Moreover, as de-coupled algorithms were proposed, we were able to study the hybridization of probabilistic first-order and deterministic second-order aspects, leading to a provably second-order convergent scheme, that could be both more efficient than and as robust as a typical deterministic implementation. We also proposed a classification of the probabilistic second-order properties that could be used in DFO and beyond. In our experiments, we observed that their practical satisfaction was a common event: this opens the possibility of better understanding the performance of optimization methods applied to nonconvex problems thanks to second-order probabilistic analysis.

The second goal of the thesis was to determine the relevance of using complexity analysis as a guidance for the design of numerically efficient methods. In a deterministic setting, we established that second-order complexity bounds could only be derived for specific instances of direct-search methods, that in return proved more robust on problems for which second-order aspects are particularly interesting to consider. We went one step further by proposing a de-coupling technique that improved the existing complexity bounds available for several second-order globally convergent frameworks, including derivative-based ones. The advantages of such strategies were shown in practice, as the dissociated parts of the algorithm provided better approximations of the relevant criteria.

When associated with the introduction of randomness, the complexity results holding with overwhelming probability were even more insightful. For direct-search methods, they enlightened the possible gain in the number of calls to the objective that an algorithm based on randomly generated directions could yield, a fact that was observed in practice. The same property was shown on a de-coupling strategy applied to a second-order convergent scheme, for which the probabilistic treatment of the first-order aspects resulted in a similar improvement over a standard, first-order algorithm. These findings are in agreement with the extensive use of complexity estimates as an insightful tool for investigating the behavior of general frameworks involving random elements.

As a result, it comes out from our study that the analysis of derivative-free algorithms, driven by complexity concerns, can provide a guidance into the design of more efficient methods. Meanwhile, the introduction of randomness induces practical enhancement of the methods while equipping those frameworks with probabilistic theoretical guarantees. The combination of those tools is helpful in connecting the gains in performance with the analytical results.

Perspectives

The work presented in this thesis can be extended in multiple directions, due to its applicability on other derivative-free methods than those presented in this thesis, and even beyond. We identify thereafter three main directions for future work.

In Chapter 3, we presented a practical direction generation technique that could require as few as two function evaluations to satisfy a desirable property with sufficient probability: such a proposal was not made for model-based methods, as the associated requirements are harder to ensure. The design a trust-region method based on a probabilistic construction of the models using considerably less than $n+1$ function evaluations in dimension n thus poses an interesting challenge.

A short-term perspective of the thesis builds on the work presented in Chapter 5: we indeed plan on proposing a general algorithmic framework tailored to any kind of linear constraints. As a follow-up, one might be interested in addressing general, nonlinear constraints. Given that typical direct-search implementations can be embedded in an augmented Lagrangian framework in a relatively straightforward manner [76, 84], it seems that such a study could be performed for several of the strategies we proposed.

One of the most promising developments of this work consists in adapting the de-coupling technique of Chapter 7 to a derivative-based algorithm, since nothing forbids it in our analysis. In fact, as second-order methods are regaining interest in the optimization community, proposing a way to include second-order aspects at a potentially lower expense seems particularly attractive, especially if those aspects are introduced in a probabilistic fashion. This perspective thus goes hand in hand with another one, related to the material of Chapter 8: the identification of second-order descent probabilistic properties that could be ensured by the use of random elements. We believe that a rigorous study could be performed using tools from probability theory to propose new approaches based on second-order features with enhanced practical performance.

Appendix A

Elements of probability theory

This appendix aims at clarifying the probability notions that are manipulated throughout the manuscript. We properly define the probability space(s) at hand, as well as the martingale properties that are at the heart of our almost-sure convergence results.

A.1 Probability space and random elements in an optimization method

Most of the following definitions are elementary notions in probability theory; we adopted the notations of Durrett [54].

Definition A.1 (σ-algebras) *Let $\mathcal{F}$ a non-empty collection of subsets of a set Ω. $\mathcal{F}$ is said to be a* ***σ-algebra*** *if it is closed under complementation and countable union. Let $A \subset \Omega$. The smallest σ-algebra containing A is called the* σ-algebra generated by A.

Definition A.2 (Probability space) *A* ***probability space*** *is a triple $(\Omega, \mathcal{F}, \mathbb{P})$ where:*

- *Ω is a set of* outcomes*;*
- *$\mathcal{F}$ is a σ-algebra called the σ-algebra of* events*;*
- *$\mathbb{P}$ is a* probability measure *on $\mathcal{F}$, i.e., a function from $\mathcal{F}$ to $[0,1]$ that is nonnegative and countably additive; in particular, one has $\mathbb{P}(\emptyset) = 0$ and $\mathbb{P}(\Omega) = 1$.*

Definition A.3 (Random variable) *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. $X : \Omega \to \mathbb{R}$ is said to be a* ***random variable*** *if*

$$\forall\, B \in \mathcal{R}, X^{-1}(B) = \{\omega \in \Omega, X(\omega) \in B\} \in \mathcal{F},$$

where $\mathcal{R}$ is the σ-algebra generated by the open sets in $\mathbb{R}$ (we say that $X \in \mathcal{F}$).
The ***σ-algebra generated by*** *$\mathcal{F}$ is the smallest σ-algebra such that $X \in \mathcal{F}$. It is denoted by $\sigma(X)$.*

An example of random variable is the **indicator function** of a element $A \in \mathcal{F}$ defined by:

$$\forall\, \omega \in \Omega, \quad 1_A(\omega) = \begin{cases} 1 & \text{if } \omega \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Random vectors and measurable maps In Definition A.3, one can replace $(\mathbb{R}, \mathcal{R})$ by any other measurable space. When one uses $(\mathbb{R}^n, \mathcal{R}_n)$, where $\mathcal{R}_n$ denotes the standard σ-algebra generated from the open sets in $\mathbb{R}^n$, X is called a *random vector*, otherwise it is called a *measurable map*. We will use the generic name **random element** to emphasize the random nature of these objects, but most of the time we will be talking about random vectors or random variables.

Definition A.4 *Let $\mathcal{A}$ be an iterative optimization algorithm assumed to rely on introducing a random element at every iteration. We define the* global probability space associated to $\mathcal{A}$ *as $\left(\Omega^{\mathcal{A}}, \sigma_\infty^{\mathcal{A}}, P^{\mathcal{A}}\right)$, where $\Omega^{\mathcal{A}}$ is the set of all possible outcomes for the sequences appearing in the algorithmic process, $\sigma_\infty^{\mathcal{A}}$ is the σ-algebra generated by the entire sequence of random elements and $P^{\mathcal{A}}$ is a probability measure; consequently, we will have:*

$$\forall\, B \in \sigma_\infty^{\mathcal{A}}, \quad \mathbb{P}(B) = P^{\mathcal{A}}[B].$$

Example A.1 *The global probability space for Algorithm 3.1 (direct search based on probabilistic descent) is $\left(\Omega^{\text{dspd}}, \sigma_\infty^{\text{dspd}}, P^{\text{dspd}}\right)$, where Ω^{dspd} is the set of all possible values for the iterates, polling sets and step sizes, $\sigma_\infty^{\text{dspd}}$ is the σ-algebra generated by the entire sequence of random polling sets, i.e.,*

$$\sigma_\infty^{\text{dspd}} \;=\; \sigma\left(\mathfrak{D}_0, \mathfrak{D}_1, \dots\right)$$

and P^{dspd} is the joint probability distribution of the product variable $\bigcup_{k=0}^{\infty} \mathfrak{D}_k$.

A.2 Conditional expectation, conditioning to the past and martingales

Most of the algorithms presented in this thesis rely on an introduction of randomness at every iteration. We are thus faced with sequences of random elements that are not independent, but rather are functions of random independent variables. Conditioning is a convenient way of expressing this situation.

Definition A.5 (Expectation) *Let X be a positive random variable in a probability space $(\Omega, \mathcal{F}, P)$; the **expectation** of X (also called the expected value or the mean), denoted by $\mathbb{E}[X]$, is defined by the following Lebesgue integral:*

$$\mathbb{E}[X] \;=\; \int_\Omega X(\omega)\, dP(\omega) \;\equiv\; \int_\Omega X\, dP.$$

For an arbitrary random variable X, the expectation exists if either its positive part $X^+ = \max\{X, 0\}$ or its negative part $X^- = \max\{-X, 0\}$ has a finite expectation. In this case, one has:

$$\mathbb{E}[X] \;=\; \mathbb{E}\left[X^+\right] - \mathbb{E}\left[X^-\right].$$

Definition A.6 (Conditional expectation and probability) *Let $(\Omega, \mathcal{F}, P)$ a probability space, $\mathcal{F}_0 \subset \mathcal{F}$ a σ-algebra and $X \in \mathcal{F}$ a random variable such that $\mathbb{E}[|X|] < \infty$. The **conditional expectation of** X **given** $\mathcal{F}_0$, denoted as $\mathbb{E}[X|\mathcal{F}_0]$, is defined by any random variable Y such that:*

i) $Y \in \mathcal{F}_0$,

ii) $\forall\, A \in \mathcal{F}_0,\ \int_A X\, dP \;=\; \int_A Y\, dP$.

Y is said to be a version of the conditional expectation.
*Let $A \in \mathcal{F}$, the **conditional probability of** A **given** $\mathcal{F}_0$ is defined as follows:*

$$\mathbb{P}(A|\mathcal{F}_0) \;=\; \mathbb{E}(1_A|\mathcal{F}_0).$$

It is possible to define conditional expectation with respect to a random variable, as a particular case of Definition A.6. We then have

$$\mathbb{E}[X|Z] = \mathbb{E}[X|\sigma(Z)], \tag{A.1}$$

where $\sigma(Z)$ is the σ-algebra generated by the variable Z.

One example of a property that can be stated using conditional expectations is Lemma 3.10, which we restate and prove below.

Lemma A.1 ([67, Lemma 4.4]) *Let $\{Z_k\}$ be a sequence of Bernoulli variables satisfying*

$$\mathbb{P}(Z_0 = 1) \;\geq\; p \quad \textit{and} \quad \mathbb{P}(Z_k = 1 \mid Z_0, \ldots, Z_{k-1}) \;\geq\; p \quad (k \geq 1), \tag{A.2}$$

for some $p \in (0,1)$. Let $\lambda \in (0, p)$; then

$$\pi_k(\lambda) = \mathbb{P}\left(\sum_{l=0}^{k-1} Z_l \leq \lambda\, k\right) \;\leq\; \exp\left[-\frac{(p-\lambda)^2}{2p} k\right]. \tag{A.3}$$

Proof. The result can be proved by standard techniques of large deviations. Let t be an arbitrary positive number. By Markov's Inequality,

$$\pi_k(\lambda) \;=\; \mathbb{P}\left(\exp\left(-t \sum_{l=0}^{k-1} Z_l\right) \geq \exp(-t\lambda k)\right) \;\leq\; \exp(t\lambda k)\, \mathbb{E}\left[\prod_{l=0}^{k-1} \mathrm{e}^{-tZ_l}\right]. \tag{A.4}$$

Now let us study $\mathbb{E}\left[\prod_{l=0}^{k-1} \mathrm{e}^{-tZ_l}\right]$. By Properties $\mathbf{G}^*$ and $\mathbf{K}^*$ of Shiryaev [106, page 216], we have

$$\mathbb{E}\left[\prod_{l=0}^{k-1} \mathrm{e}^{-t\,Z_l}\right] \;=\; \mathbb{E}\left[\mathbb{E}\left[\mathrm{e}^{-t\,Z_{k-1}} \mid Z_0, Z_1, \ldots, Z_{k-2}\right] \prod_{l=0}^{k-2} \mathrm{e}^{-t\,Z_l}\right]. \tag{A.5}$$

According to (A.2) and the fact that the function $r\,\mathrm{e}^{-t} + (1-r)$ is monotonically decreasing in r, it holds (with $\bar{p} = \mathbb{P}\left(Z_{k-1} = 1 \mid Z_0, Z_1, \ldots, Z_{k-2}\right) \geq p$)

$$\mathbb{E}\left[\mathrm{e}^{-t\,Z_{k-1}} \mid Z_0, Z_1, \ldots, Z_{k-2}\right] \;=\; \bar{p}\,\mathrm{e}^{-t} + (1-\bar{p}) \;\leq\; p\,\mathrm{e}^{-t} + (1-p) \;\leq\; \exp\Big(p\,\mathrm{e}^{-t} - p\Big)\,,$$

which implies, from equality (A.5), that

$$\mathbb{E}\left[\prod_{l=0}^{k-1} \mathrm{e}^{-t\,Z_l}\right] \leq \exp\Big(p\,\mathrm{e}^{-t} - p\Big)\,\mathbb{E}\left[\prod_{l=0}^{k-2} \mathrm{e}^{-t\,Z_l}\right].$$

By recursively iterating the above estimation, we finally arrive at

$$\mathbb{E}\left[\prod_{l=0}^{k-1} \mathrm{e}^{-t\,Z_l}\right] \;\leq\; \exp\Big[k\,(p\,\mathrm{e}^{-t} - p)\Big]\,.$$

Inequality (A.4) can then be rewritten as

$$\pi_k(\lambda) \;\leq\; \exp\Big[k\,(t\,\lambda + p\,\mathrm{e}^{-t} - p)\Big]\,, \tag{A.6}$$

which holds for all $t > 0$. Let us select $t = \ln(\lambda^{-1}\,p)$. Then we have

$$t\,\lambda + p\,\mathrm{e}^{-t} - p \;=\; \lambda\,\ln(\lambda^{-1}\,p) + \lambda - p \;=\; -\frac{1}{2\,\xi}(\lambda - p)^2 \qquad (\lambda < \xi < p),$$

the second equality coming from Taylor expansion of the function $\lambda \mapsto \lambda \ln(\lambda^{-1}p) + \lambda - p$ at the point p. Thus, we conclude from inequality (A.6) that

$$\pi_k(\lambda) \;\leq\; \exp\left[-\frac{(\lambda - p)^2}{2\,p}k\right].$$

□

Definition A.7 (Submartingale) *Let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $\{\mathcal{F}_k\}_k$ be a fibration, i.e., a sequence of σ-algebras such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$. A sequence $\{X_k\}_k$ of random variables is a* ***submartingale*** *if:*

i) $\forall\, k, X_k \in \mathcal{F}_k$ *($\{X_k\}$ is said to be adapted to $\{\mathcal{F}_k\}$),*

ii) $\forall\, k, \mathbb{E}\left[|X_k|\right] \;<\; \infty$,

iii) $\forall\, k, \mathbb{E}\left[X_{k+1}|\mathcal{F}_k\right] \;\geq\; X_k$.

Submartingales have favorable asymptotic properties as shown by the following theorem (for a proof, see [54, Theorem 5.3.1.]).

Theorem A.1 (Convergence of submartingales with bounded increments) *Let X_k be a submartingale with $|X_k - X_{k-1}| \leq M < \infty$ for some $M \geq 0$. Then, with probability one, either $\lim_{k\to\infty} X_k$ exists and is finite, or $\limsup_{k\to\infty} X_k = \infty$.*

With the help of Theorem A.1, we can prove Lemma 3.5, which is restated as Lemma A.2. Note that it is a mild generalization of [14, Lemma 4.1] that considers two parameters $\theta \in (0,1)$ and $\gamma > 1$ instead of one parameter $\gamma > 1$ and its inverse.

Lemma A.2 ([14, Lemma 4.1]) *Let $\theta \in (0,1)$ and $\gamma > 1$. Let also $\{Y_k\}$ be a sequence of nonnegative uniformly bounded random variables, and $\{\Gamma_k\}$ a sequence of Bernoulli random variables taking the values $\ln\gamma$ and $\ln\theta$, such that*

$$\mathbb{P}\left[\Gamma_k = \ln\gamma |\ \sigma(\Gamma_0,\ldots,\Gamma_{k-1}), \sigma(Y_0,\ldots,Y_k)\right] \ \geq\ \frac{\ln\theta}{\ln(\gamma^{-1}\,\theta)}.$$

We define P as the set of indexes k such that $\Gamma_k = \ln\gamma$ and $\mathcal{N} = \mathbb{N} \setminus P$. Then

$$\mathbb{P}\left[\left\{\sum_{i\in P} Y_i < \infty\right\} \bigcap \left\{\sum_{i\in\mathcal{N}} Y_i = \infty\right\}\right] \ =\ 0.$$

Proof. Consider the sequence of random variables $\{\Lambda_k\}$ defined by $\Lambda_k = \Lambda_{k-1} + \Gamma_k\, Y_k$. One can show that this is a submartingale of bounded increments $\{\Gamma_k\, Y_k\}$; we thus know that $\mathbb{P}\left[\{\limsup_k \Lambda_k = -\infty\}\right] = 0$.
Besides, considering that

$$\Lambda_k = \ln\gamma \sum_{\substack{i\in P\\ i\leq k}} Y_i + \ln\theta \sum_{\substack{i\in\mathcal{N}\\ i\leq k}} Y_i,$$

we have that

$$\left\{\sum_{i\in P} Y_i < \infty\right\} \bigcap \left\{\sum_{i\in\mathcal{N}} Y_i = \infty\right\} \subset \left\{\limsup_k \Lambda_k = -\infty\right\}.$$

This gives us the desired result. □

Appendix B

List of CUTEst test problems

B.1 Nonconvex test problems

Table B.1 lists all the nonconvex problems used in this thesis for establishing performance profiles. The standard benchmark of problems used in Chapter 6 and Section 7.5.2 contained all problems except for MEYER3. In Section 7.5.1, problem MEYER3 was used but problems BEALE, GROWTHLS, HIELOW, INDEF, SCOSINE and STRATEC removed from the list.

B.2 Linearly-constrained test problems

Table B.2 lists the bound-constrained problems tested in Section 5.4.1, while Table B.3 corresponds to the test problems from Section 5.4.2, with linear equality constraints and possibly bounds on the variables.

Problem	Size	Problem	Size	Problem	Size
ALLINITU	4	BARD	3	BEALE	2
BIGGS6	6	BOX3	3	BROWNAL	10
BROYDN7D	10	BRYBND	10	CHNROSNB	10
DENSCHND	3	DENSCHNE	3	DIXMAANA	15
DIXMAANB	15	DIXMAANC	15	DIXMAAND	15
DIXMAANE	15	DIXMAANF	15	DIXMAANG	15
DIXMAANH	15	DIXMAANI	15	DIXMAANJ	15
DIXMAANK	15	DIXMAANL	15	ENGVAL2	3
ERRINROS	10	EXPFIT	2	FMINSURF	16
FREUROTH	10	GROWTHLS	3	GULF	3
HAIRY	2	HATFLDD	3	HATFLDE	3
HEART6LS	6	HEART8LS	8	HELIX	3
HIELOW	3	HIMMELBB	2	HIMMELBG	2
HUMPS	2	INDEF	10	KOWOSB	4
LOGHAIRY	2	MANCINO	10	MARATOSB	2
MEYER	3	MSQRTALS	4	MSQRTBLS	9
OSBORNEA	5	OSBORNEB	11	PENALTY3	50
SCOSINE	10	SINQUAD	5	SNAIL	2
SPARSINE	10	SPMSRTLS	28	STRATEC	10
VAREIGVL	10	VIBRBEAM	8	WOODS	4
YFITU	3				

Table B.1: Nonconvex test problems from `CUTEst`.

Problem	Size	Bounds	Problem	Size	Bounds	Problem	Size	Bounds
ALLINIT	4	5	BQP1VAR	1	2	CAMEL6	2	4
CHEBYQAD	20	40	CHENHARK	10	10	CVXBQP1	10	20
DEGDIAG	11	11	DEGTRID	11	11	DEGTRID2	11	11
EG1	3	4	EXPLIN	12	24	EXPLIN2	12	24
EXPQUAD	12	12	HARKERP2	10	10	HART6	6	12
HATFLDA	4	4	HATFLDB	4	5	HIMMELP1	2	4
HS1	2	1	HS25	3	6	HS2	2	1
HS38	4	8	HS3	2	1	HS3MOD	2	1
HS45	5	10	HS4	2	2	HS5	2	4
HS110	10	20	JNLBRNG1	16	28	JNLBRNG2	16	28
JNLBRNGA	16	28	KOEBHELB	3	2	LINVERSE	19	10
LOGROS	2	2	MAXLIKA	8	16	MCCORMCK	10	20
MDHOLE	2	1	NCVXBQP1	10	20	NCVXBQP2	10	20
NCVXBQP3	10	20	NOBNDTOR	16	32	OBSTCLAE	16	32
OBSTCLBL	16	32	OSLBQP	8	11	PALMER1A	6	2
PALMER2B	4	2	PALMER3E	8	1	PALMER4A	6	2
PALMER5B	9	2	PFIT1LS	3	1	POWELLBC	20	40
PROBPENL	10	20	PSPDOC	4	1	QRTQUAD	12	12
S368	8	16	SCOND1LS	12	24	SIMBQP	2	2
SINEALI	20	40	SPECAN	9	18	TORSION1	16	32
TORSIONA	16	32	WEEDS	3	4	YFIT	3	1

Table B.2: Bound-constrained test problems from `CUTEst`.

Problem	Size	Bounds	Lin. Eq	Problem	Size	Bounds	Lin. Eq.
BT3	5	0	3	HIMMELBA	2	0	2
HS9	2	0	1	HS28	3	0	1
HS48	5	0	2	HS49	5	0	2
HS50	5	0	3	HS51	5	0	3
HS52	5	0	3	ZANGWIL3	3	0	3
CVXQP1	10	20	5	CVXQP2	10	20	2
DEGTRIDL	11	11	1	FERRISDC	16	24	7
GOULDQP1	32	64	17	HONG	4	8	1
HS41	4	8	1	HS53	5	10	3
HS54	6	12	1	HS55	6	8	6
HS62	3	6	1	HS112	10	10	3
NCVXQP1	10	20	5	NCVXQP2	10	20	5
NCVXQP3	10	20	5	NCVXQP4	10	20	2
NCVXQP5	10	20	2	NCVXQP6	10	20	2
PORTFL1	12	24	1	PORTFL2	12	24	1
PORTFL3	12	24	1	PORTFL4	12	24	1
PORTFL6	12	24	1	PORTSNQP	10	10	2
PORTSQP	10	10	1	READING2	9	14	4
SOSQP1	20	40	11	SOSQP2	20	40	11
STCQP1	17	34	8	STCQP2	17	34	8
STNQP1	17	34	8	STNQP2	17	34	8
TAME	2	2	1	TWOD	31	62	10

Table B.3: Linearly-constrained test problems (linear equalities and possibly bounds) from `CUTEst`.

Bibliography

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards, United States Department of Commerce, Washington, tenth edition, 1972.

[2] M. A. Abramson. Second-order behavior of pattern search. *SIAM J. Optim.*, 16:315–330, 2005.

[3] M. A. Abramson and C. Audet. Convergence of mesh adaptive direct search to second-order stationarity points. *SIAM J. Optim.*, 17:606–619, 2006.

[4] M. A. Abramson, C. Audet, J. E. Dennis Jr., and S. Le Digabel. OrthoMADS: A deterministic MADS instance with orthogonal directions. *SIAM J. Optim.*, 20:948–966, 2009.

[5] M. A. Abramson, O. A. Brezhneva, J. E. Dennis Jr., and R. L. Pingel. Pattern search in the presence of degenerate linear constraints. *Optim. Methods Softw.*, 23:297–319, 2008.

[6] M. A. Abramson, L. Frimannslund, and T. Steihaug. A subclass of generating set search with convergence to second-order stationary points. *Optim. Methods Softw.*, 29:900–918, 2014.

[7] E. Artin. *The Gamma Function*. Holt, Rinehart and Winston, New York, 1964. Translated to English by M. Butler.

[8] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217, 2006.

[9] C. Audet, S. Le Digabel, and M. Peyrega. Linear equalities in blackbox optimization. *Comput. Optim. Appl.*, 61:1–23, 2015.

[10] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM J. Optim.*, 13:889–903, 2003.

[11] A. Auslender. Computing points that satisfy second order necessary optimality conditions for unconstrained minimization. *SIAM J. Optim.*, 20:1868–1884, 2010.

[12] C. P. Avelino, J. M. Moguerza, A. Olivares, and F. J. Prieto. Combining and scaling descent and negative curvature directions. *Math. Program.*, 128:285–319, 2011.

[13] 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.

[14] 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.

[15] E. Bergou, S. Gratton, and L. N. Vicente. Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation. *SIAM/ASA J. Uncertain. Quantif.*, 4:924–951, 2016.

[16] Á. Bűrmen, J. Olenšek, and T. Tuma. Mesh adaptive direct search with second directional derivative-based Hessian update. *Comput. Optim. Appl.*, 62:693–715, 2015.

[17] E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos, and Ph. L. Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Math. Program.*, 2016. doi: 10.1007/s10107-016-1065-8.

[18] E. G. Boman. *Infeasibility and Negative Curvature in Optimization.* PhD thesis, Stanford University, February 1999.

[19] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. arXiv, 1606.04838, 2016.

[20] R. H. Byrd, R. B. Schnabel, and G. A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM J. Numer. Anal.*, 24:1152–1170, 1987.

[21] G. Capasso. A deterministic method for the multiobjective optimization of electromagnetic devices and its application to pose detection for magnetic-assisted medical applications. Master's thesis, Dipartimento Di Ingegneria Industriale, Università degli Studi di Padova, Padova, Italy, 2015.

[22] 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.

[23] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130:295–319, 2011.

[24] C. Cartis, N. I. M. Gould, and Ph. L. Toint. Optimal Newton-type methods for nonconvex optimization. Technical Report naXys-17-2011, Dept of Mathematics, FUNDP, Namur (B), 2011.

[25] 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.

[26] 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.

[27] C. Cartis, N. I. M. Gould, and Ph. L. Toint. An example of slow convergence for Newton's method on a function with globally Lipschitz continuous Hessian. Technical Report ERGO 13-008, School of Mathematics, Edinburgh University, 2013.

[28] 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.

[29] C. Cartis and K. Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. arXiv, 1505.06070v1, 2015.

[30] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. arXiv, 1504.04231v1, 2015.

[31] Z. Chen and J. J. Dongarra. Condition numbers of Gaussian random matrices. *SIAM J. Matrix Anal. Appl.*, 27:603–620, 2005.

[32] E. Çınlar. *Probability and Stochastics*. Graduate Texts in Mathematics. Springer, New York, 2011.

[33] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

[34] A. R. Conn, K. Scheinberg, and Ph. L. Toint. A derivative free optimization algorithm in practice. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St Louis, MO, 1998.

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

[36] 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.

[37] 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.

[38] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[39] I. D. Coope and C. J. Price. A direct search conjugate directions algorithm for unconstrained minimization. *ANZIAM J.*, 42:C478–C498, 2000.

[40] 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.*, 2016. doi:10.1007/s10107-016-1026-2.

[41] A. L. Custódio. *Aplicações de Derivadas Simplécticas em Métodos de Procura Directa*. PhD thesis, Universidade Nova de Lisboa, 2007.

[42] A. L. Custódio and L. N. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM J. Optim.*, 18:537–555, 2007.

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

[44] J. E. Dennis Jr. and V. Torczon. Direct search methods on parallel machines. *SIAM J. Optim.*, 1:448–474, 1991.

[45] J. E. Dennis Jr. and L. N. Vicente. On the convergence theory of trust-region-based algorithms for equality-constrained optimization. *SIAM J. Optim.*, 7:927–950, 1997.

[46] Y. Diouane, S. Gratton, and L. N. Vicente. Globally convergent evolution strategies for constrained optimization. *Comput. Optim. Appl.*, 62:323–346, 2015.

[47] M. Dodangeh and L. N. Vicente. Worst case complexity of direct search under convexity. *Math. Program.*, 155:307–332, 2016.

[48] 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.

[49] B. Doerr. Analyzing randomized search heuristics: Tools from probability theory. In A. Auger and B. Doerr, editors, *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, volume 1 of *Theoret. Comput. Sci.*, pages 1–20. World Scientific, Singapore, 2011.

[50] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[51] D. W. Dreisigmeyer. Equality constraints, Riemannian manifolds and direct-search methods. Technical Report LA-UR-06-7406, Los Alamos National Laboratory, 2006.

[52] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, Cambridge, 2009.

[53] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: the power of two function evaluations. *IEEE Trans. Inform. Theory*, 61:2788–2806, 2015.

[54] R. Durrett. *Probability: Theory and Examples.* Camb. Ser. Stat. Prob. Math. Cambridge University Press, Cambridge, fourth edition, 2010.

[55] G. Fasano, J. L. Morales, and J. Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optim. Methods Softw.*, 24:145–154, 2009.

[56] A. Forsgren and W. Murray. Newton methods for large-scale linear inequality-constrained minimization. *SIAM J. Optim.*, 7:162–176, 1997.

[57] L. Frimannslund and T. Steihaug. A generating set search method using curvature information. *Comput. Optim. Appl.*, 38:105–121, 2007.

[58] U. M. García-Palomares, I. J. García-Urrea, and P. S. Rodríguez-Hernández. On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization. *Optim. Methods Softw.*, 28:1233–1261, 2013.

[59] U. M. García-Palomares and J. F. Rodríguez. New sequential and parallel derivative-free algorithms for unconstrained minimization. *SIAM J. Optim.*, 13:79–96, 2002.

[60] 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.

[61] R. Garmanjani and L. N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, 33:1008–1028, 2013.

[62] D. Goldfarb. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. *Math. Program.*, 18:31–40, 1980.

[63] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. *Optim. Methods Softw.*, 14:75–98, 2000.

[64] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEr, a Constrained and Unconstrained Testing Environment, revisited. *ACM Trans. Math. Software*, 29:373–394, 2003.

[65] 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.

[66] 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.

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

[68] S. Gratton, A. Sartenaer, and Ph. L. Toint. Second-order convergence properties of trust-region methods using incomplete curvature information, with an application to multigrid optimization. *J. Comput. Math.*, 24:676–692, 2006.

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

[70] L. Grippo, F. Lampariello, and S. Lucidi. Global convergence and stabilization of unconstrained minimization methods. *J. Optim. Theory Appl.*, 56:385–406, 1988.

[71] G. Hillier. The density of a quadratic form in a vector uniformly distributed on the n-sphere. *Econometric Theory*, 17:1–28, 2001.

[72] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419–426, 1961.

[73] D. Júdice. *Trust-Region Methods without using Derivatives: Worst Case Complexity and the Non-smooth Case.* PhD thesis, Dept. Mathematics, Univ. Coimbra, 2015.

[74] C. T. Kelley. *Implicit Filtering.* Software Environment and Tools. SIAM, Philadelphia, 2011.

[75] 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.

[76] T. G. Kolda, R. M. Lewis, and V. Torczon. A generating set direct search augmented lagrangian algorithm for optimization with a combination of general and linear constraints. Technical Report SAND2006-5315, SANDIA National Laboratories, 2006.

[77] T. G. Kolda, R. M. Lewis, and V. Torczon. Stationarity results for generating set search for linearly constrained optimization. *SIAM J. Optim.*, 17:943–968, 2006.

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

[79] J. Larson and S. C. Billups. Stochastic derivative-free optimization using a trust region framework. *Comput. Optim. Appl.*, 64:619–645, 2016.

[80] R. M. Lewis, A. Shepherd, and V. Torczon. Implementing generating set search methods for linearly constrained minimization. *SIAM J. Sci. Comput.*, 29:2507–2530, 2007.

[81] R. M. Lewis and V. Torczon. Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.*, 9:1082–1099, 1999.

[82] R. M. Lewis and V. Torczon. Pattern search algorithms for linearly constrained minimization. *SIAM J. Optim.*, 10:917–941, 2000.

[83] R. M. Lewis and V. Torczon. Active set identification for linearly constrained minimization without derivatives. *SIAM J. Optim.*, 20:1378–1405, 2009.

[84] R. M. Lewis and V. Torczon. A direct search approach to nonlinear programming problems using an augmented Lagrangian method with explicit treatment of the linear constraints. Technical Report WM-CS-2010-01, College of William & Mary, Department of Computer Science, 2010.

[85] L. Liu and X. Zhang. Generalized pattern search methods for linearly equality constrained optimization problems. *Appl. Math. Comput.*, 181:527–535, 2006.

[86] S. Lucidi, F. Rochetich, and M. Roma. Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization. *SIAM J. Optim.*, 8:916–939, 1998.

[87] S. Lucidi and M. Sciandrone. A derivative-free algorithm for bound constrained minimization. *Comput. Optim. Appl.*, 21:119–142, 2002.

[88] S. Lucidi and M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM J. Optim.*, 13:97–116, 2002.

[89] The Mathworks, Inc. *Global Optimization Toolbox User's Guide,version 3.3*, October 2014.

[90] The Mathworks, Inc. *MATLAB R2016a, Trial Version*, February 2016.

[91] R. Mifflin. A superlinearly convergent algorithm for minimization without evaluating derivatives. *Math. Program.*, 9:100–117, 1975.

[92] A. A. Mohsenipour. *On the distribution of quadratic expressions in various types of random vectors*. PhD thesis, The University of Western Ontario, December 2012.

[93] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Comput.*, 4:553–572, 1983.

[94] J.J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Program.*, 16:1–20, 1979.

[95] J.J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, 20:172–191, 2009.

[96] J.-J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires. *Comptes Rendus de l'Académie des Sciences de Paris*, 255:238–240, 1962.

[97] M. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2:19–20, 1959.

[98] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, pages 308–313, 1965.

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

[100] Yu. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22:341–362, 2012.

[101] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, second edition, 2006.

[102] A. Olivares, J. M. Moguerza, and F. J. Prieto. Nonconvex optimization using negative curvature within a modified linesearch. *European J. Oper. Res.*, 189:706–722, 2008.

[103] M. J. D. Powell. Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. *Math. Program.*, 100:183–215, 2004.

[104] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Program.*, 144:1–38, 2014.

[105] K. Scheinberg and Ph. L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM J. Optim.*, 20:3512–3532, 2010.

[106] A. N. Shiryaev. *Probability*. Grad. Texts on Math. Springer-Verlag, New York, 1995.

[107] G. A. Shultz, R. B. Schnabel, and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985.

[108] J. C. Spall. *Introduction to stochastic search and optimization: Estimation, simulation and control*. Wiley-Interscience. John Wiley & Sons, Hoboken, New Jersey, 2003.

[109] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7:1–25, 1997.

[110] P. Tseng. Fortified-descent simplicial search method: A general approach. *SIAM J. Optim.*, 10:269–288, 1999.

[111] K. Ueda and N. Yamashita. On a global complexity bound of the Levenberg-Marquardt method. *J. Optim. Theory Appl.*, 147:443–453, 2010.

[112] A. I. F. Vaz and L. N. Vicente. A particle swarm pattern search method for bound constrained global optimization. *J. Global Optim.*, 39:197–219, 2007.

[113] A. I. F. Vaz and L. N. Vicente. PSwarm: A hybrid solver for linearly constrained global derivative-free optimization. *Optim. Methods Softw.*, 24:669–685, 2009.

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

[115] L. N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Math. Program.*, 133:299–325, 2012.

[116] A. A. Zhigljavsky and A. G. Zilinskas. *Stochastic Global Optimization*. Springer-Verlag, Berlin, 2008.