SAPIENZA
UNIVERSITÀ DI ROMA

Derivative-Free Optimization: worst-case complexity for Line-Search methods and a Mixed Penalty-Barrier approach.

Department of computer, control and management engineering ANTONIO RUBERTI (DIAG)
Automatic Control, Bioengineering and Operations Research (ABRO)
(XXXII cycle)

Andrea Brilli
ID number 1708157

Advisor
Prof. Giampaolo Liuzzi

Director
Prof. Laura Palagi

Academic Year 2023/2024

Derivative-Free Optimization: worst-case complexity for Line-Search methods and a Mixed Penalty-Barrier approach.
PhD thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: brilli@diag.uniroma1.it

Acknowledgments

This thesis would not have been possible without the guidance, support, and encouragement of many individuals. I would like to take this opportunity to express my gratitude to them.

First and foremost, I would like to express my deepest gratitude to my PhD supervisor, Giampaolo Liuzzi, for his unwavering support, guidance, and patience throughout this journey. Giampaolo has always been available, making time to discuss my ideas, no matter how complex or unclear they may have seemed at first. His ability to listen to my requests and provide thoughtful, practical advice has been invaluable in overcoming the many challenges I faced during these years. Most of all, I am profoundly grateful for the way he treated me—not just as a student, but as a peer and a collaborating researcher. This approach has profoundly shaped my ability to work independently and built my confidence in pursuing and trusting my own ideas.

I would also like to extend a heartfelt thank you to my Master's thesis advisor, Stefano Lucidi, who has been an integral part of this journey alongside my PhD supervisor. Stefano's openness to my ideas, regardless of how well-formed they were, has deeply inspired me to adopt the same attitude when working and collaborating with others. His encouragement and belief in me have taught me resilience and perseverance, even when faced with seemingly insurmountable challenges. The lessons I have learned from him will remain a guiding light in my future endeavors.

During my PhD, I was fortunate to have two visiting research periods, each of which left a lasting impact on my academic and personal growth. I would like to thank Professor Ana Luísa Custódio from Nova University in Lisbon, and Professors Sébastien Le Digabel and Youssef Diouane from Polytechnique Montréal, for their warm welcome and for making me feel at home in their institutions. Their genuine interest in knowing me not just professionally but personally as well was deeply meaningful. The esteem they consistently showed for me, combined with their generosity in sharing their knowledge and expertise, has been instrumental in my growth as a researcher and as an individual.

A special thank you goes to Everton Silva, a PhD student supervised by Professor Custódio, with whom I started collaborating during my time in Lisbon. Since we began working together, our collaboration has never stopped, and it has been an exciting and rewarding experience to plan and pursue new research directions alongside him. Sharing this journey with Everton has been incredibly important in keeping my motivation high and making this academic path much more fulfilling.

I am also deeply thankful to all my colleagues and lab mates from the three institutions where I worked during this PhD: Sapienza University of Rome, Nova University of Lisbon, and Polytechnique Montréal. The exchanges, discussions, and shared moments I experienced with all of you have greatly enriched this journey and created lasting memories.

I would like to extend my gratitude to the revisors of this thesis, whose insightful suggestions and constructive feedback have significantly improved its quality.

Finally, I would like to thank my family for their unwavering support, encouragement, and love throughout this journey. Your belief in me has been a constant source of strength, and I would not have reached this point without you.

This work is as much a reflection of your support and encouragement as it is of my efforts. Thank you all.

Abstract

This thesis investigates new advances in Derivative-Free Optimization (DFO), focusing on complexity analysis for line-search methods and proposing a novel mixed penalty-barrier approach for handling constraints in black-box optimization problems. Classical optimization approaches often rely on gradient information; however, in practical applications, derivatives may be unavailable, unreliable, or costly to compute. DFO addresses this gap by developing methods that depend solely on function evaluations, which are crucial for applications with noisy, expensive, or simulation-based objective functions.

Chapter 2 presents a theoretical analysis of line-search based DFO algorithms, concentrating on their worst-case complexity bounds in both unconstrained and bound-constrained contexts. For unconstrained problems, a worst-case complexity bound is established that matches the iteration requirements of existing direct search methods, marking a significant theoretical contribution to the field. Additionally, for the bound-constrained case, a criticality measure is introduced to evaluate solution quality, with complexity bounds developed to ensure efficient progress. An active-set identification property is also demonstrated, showing that the method can recognize and exploit bounds that are active at the solution, enhancing computational efficiency for bound-constrained problems.

Chapter 3 introduces a sequential mixed-penalty approach for solving general nonlinear constrained black-box optimization problems, where traditional derivative-based constraint-handling techniques are unsuitable. This approach combines two distinct penalty mechanisms: a logarithmic barrier for handling inequality constraints and an exterior penalty for equality constraints. This dual strategy leverages a line-search framework to enforce constraint satisfaction while accommodating variable bounds, demonstrating improved feasibility attainment in constrained black-box settings. Furthermore, a direct search variant of the algorithm is developed, incorporating the mixed penalty strategy to manage unrelaxable constraints effectively.

Together, these contributions advance the field of DFO by providing rigorous complexity bounds for line-search methods and a robust mixed penalty-barrier framework for constrained optimization. Empirical tests validate the efficacy of these algorithms across a diverse range of constrained optimization problems, underscoring their applicability in real-world black-box settings.

Contents

Notation

Throughout the thesis, vectors will be written in lowercase boldface (e.g., $\mathbf{v} \in \mathbb{R}^n$, $n \geq 2$), and the i-th component of $\mathbf{v} \in \mathbb{R}^n$ will be denoted as $(\mathbf{v})_i$ or v_i in absence of ambiguity. Matrices will be written in uppercase boldface (e.g., $\mathbf{S} \in \mathbb{R}^{n \times p}$). The set of column vectors of a matrix $\mathbf{D}$ will be denoted by $\mathcal{D}$, and more generally sets such as $\mathbb{N}, \mathbb{Q}, \mathbb{R}$ will be denoted by blackboard letters. The set of nonnegative real numbers will be denoted by $\mathbb{R}_+$. Sequences indexed by $\mathbb{N}$ will be denoted by $\{a_k\}_{k \in \mathbb{N}}$ or $\{a_k\}$ in absence of ambiguity. Let $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_p\} \subset \mathbb{R}^n$ be a (finite) set of vectors, we denote as $\text{cone}(\mathcal{A})$ the cone generated by the positive combinations of the vectors $\mathbf{a}_i$. Given a point $\mathbf{x} \in \mathbb{R}^n$ and a set $\Omega \subset \mathbb{R}^n$, we denote by $\mathbf{x}_\Omega$ the projection of $\mathbf{x}$ onto Ω, we use the notation $T_\Omega(\mathbf{x})$ to denote the tangent cone to Ω at $\mathbf{x}$ and $G_{T_\Omega(\mathbf{x})}$ to denote the set of generators for $T_\Omega(\mathbf{x})$, and we use $N_\Omega(\mathbf{x})$ to denote the normal cone to Ω at $\mathbf{x}$. Given $\beta \in \mathbb{R}$, we indicate the sign of β by $\text{sign}(\beta)$, that is, $\text{sign}(\beta)$ is -1 if $\beta < 0$, 0 if $\beta = 0$ and 1 if $\beta > 0$. Finally, $\|\mathbf{v}\|$ denotes the ℓ_2-norm of vector $\mathbf{v}$.

Chapter 1

Introduction

Optimization plays a crucial role in almost every discipline where decision-making is based on improving the performance of a system. From minimizing costs in logistics and operations to maximizing efficiency in engineering designs and machine learning models, the ability to find optimal solutions is a foundational task in applied mathematics and computational science. Classical optimization techniques rely heavily on gradient-based methods, which exploit derivative information to guide the search for an optimal solution. However, in many real-world scenarios, these derivatives are either unavailable or too costly to compute, necessitating alternative approaches.

When derivatives cannot be accessed directly, either due to the complexity of the objective function or the nature of the problem itself, Derivative-Free Optimization (DFO) becomes a critical tool. DFO refers to a class of optimization techniques designed to solve problems where gradient information is not available, unreliable, or impractical to compute. These methods rely solely on the values of the objective function to search for the best solution.

DFO has become increasingly relevant due to the growing number of applications where the objective functions are either noisy, discontinuous, non-smooth, or computationally expensive. For instance, in engineering design, optimizing the shape of an aircraft wing often requires running complex fluid dynamics simulations, where derivative information may be difficult or impossible to compute. Similarly, in machine learning, hyperparameter tuning tasks can be modeled as black-box optimization problems, where the evaluation of model performance is possible, but obtaining derivatives with respect to hyperparameters is not feasible. In such cases, DFO methods provide a practical approach to identifying high-quality solutions without relying on derivatives.

This thesis is situated within the field of DFO and focuses on developing and analyzing optimization algorithms tailored to specific classes of problems. The problems under consideration are characterized by computationally expensive objective functions, where each function evaluation is costly in terms of time or computational resources. Additionally, these problems often involve constraints, either in the form of equality or inequality conditions, further complicating the optimization process. The primary objective of this thesis is to provide new theoretical results on the efficiency of line-search methods, and propose a technique that can efficiently handle constrained optimization problems within the DFO framework.

The first section of this chapter provides a detailed background on DFO, highlighting the key principles and methodologies that form the foundation of this field. Particular attention is given to the classes of algorithms that are most relevant to the research presented in this thesis. These include line-search techniques and direct search methods.

The second section shifts focus to nonlinear constrained optimization problems, which are frequently encountered in real-world applications. These problems involve finding the minimum or maximum of an objective function subject to constraints on the decision variables. Constraints may take the form of equality conditions (e.g., physical laws or conservation requirements) or inequality conditions (e.g., safety limits or operational thresholds). The section introduces the fundamental concepts required to understand such problems, along with a brief description of the principal techniques used to solve them.

By combining the theoretical foundations of DFO with a practical understanding of constrained optimization problems, this thesis seeks to make significant contributions to the development of algorithms capable of solving these challenging problems more efficiently. The research presented in this thesis addresses the pressing need for optimization techniques that are not only flexible in their application but also scalable and efficient in the context of expensive and constrained functions.

In summary, this introduction chapter sets the stage for the two core aspects of the thesis: first, the exploration of DFO techniques as they relate to expensive optimization problems, and second, the integration of these techniques into the broader context of constrained optimization. The following sections will delve deeper into these topics, outlining the key algorithms and methodologies that serve as the foundation for the research conducted.

1.1 Derivative Free Optimization

What is DFO? In [6] C. Audet and W. Hare write:

> Derivative-free optimization (DFO) is the mathematical study of optimization algorithms that do not use derivatives.

DFO has been an active field of research for many decades. It arises from problems where the analytical expression of the objective function is not available nor reliable. The field of research focused on solving the latter class of problems is known as black-box optimization (BBO). In mathematical optimization, a black-box is thought of as an input-output process wherein the patterns defining the relation between input and output are not explicitly known, so that analytical information about the underlying functions is not available.

The algorithm forming the basis of large part of modern DFO methods is the *Coordinate Search* (CS) [44], also known as *Compass Search.* CS aims to solve the unconstrained problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}),$$

where $f : \mathbb{R}^n \to \mathbb{R}$. Starting from an initial point $\mathbf{x}_0$, the method iterates over the coordinate directions $\mathbf{e}_i$. At each iteration k it performs a step of length α_k along each $\mathbf{e}_i$ and it searches for a new candidate solution improving the value of the objective function, i.e. $f(\mathbf{x}_k + \alpha_k \mathbf{e}_i) < f(\mathbf{x})$. If it fails to find such a point, the step-size α_k is reduced by a factor, otherwise the step-size keeps the same value. The method is intuitive and practical to implement, which is one of the reasons for its popularity. Even though the authors did not provide theoretical guarantees for the algorithm to converge towards stationary points, it has been shown later on by Polak [82] that, provided that f is continuously differentiable and a global solution of the problem exists, the method does eventually converge to points where the gradient of the objective function is equal to 0.

We follow the classification in [6] to introduce the main classes of DFO methods proposed in literature

- *Genetic Algorithm* (GA). These methods are called *population-based.* Starting from an initial population of candidate solutions, they exploit a so-called *fitness function* to assess the quality of each candidate. Based on their fitness values, the candidates are then recombined to provide a new generation population from which the process starts all over.

- *Model-based* (MB). These methods generate a set of points with suitable geometric properties. Afterwards, the points are used to build a local model of the objective function. The model is then exploited to identify a *descent direction*, or it is minimized within a *trust-region* to identify a new candidate solution.
- *Direct Search* (DS). These methods follow the same structure of CS. The key difference is that DS algorithms can generate a different set of directions at each iteration. Furthermore, they allow for heuristic strategies to be incorporated within an additional step.
- *Line-search* (LS). Similarly to CS and DS methods, LS algorithms iterate over a set of directions, performing steps on each one to try and improve the value of the objective function. In this case there are two key differences. First, they exploit a different step-size for each direction. Furthermore, they use the notion of *sufficient decrease* to define a successful decrease of the objective function, and, when such an improvement is found over any direction, the step-size is increased as long as the objective keeps decreasing.

Note that GA aims to reach a global solution of the problem, whereas the other classes of methods have theoretical guarantees to reach general stationary points. In the results proposed in this thesis we are not concerned with algorithms belonging to the classes of GA and MB. For further understanding about these methods, more details and references can be found in [6, 30].

1.1.1 Direct Search Methods

The term *direct search* appears for the first time in the work by Hooke and Jeeves in 1961 [84]. They provide the following definition:

> We use the phrase “direct search” to describe sequential examination of trial solutions involving comparison of each trial solution with the “best” obtained up to that time together with a strategy for determining (as a function of earlier results) what the next trial solution will be. The phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques of classical analysis except where there is a demonstrable advantage in doing so.

In more recent years, the latter definition is used by R. M. Lewis et al. in [69] to discuss what DS methods are and are not. They stress on how *"derivative-free"* or *"zero-th order"* are not suitable definitions. The authors point out that even though it is true that such methods do not use derivative, and that they can be seen as zero-th order in the terms of Taylor expansions, i.e. they only use values of the objective function, this is not enough to distinguish them from other optimization methods. There exist many algorithms wherein only values of the objective function are used, such as finite differences or MB methods, but they still rely on the approximation of, at least, first-order information. In the past two decades it has been shown how exploiting surrogate models of the objective function can enhance the performance of DS methods, and it has become a key step of more sophisticated modern algorithms. Nevertheless, such strategies are only used as heuristics and they do not play any role in the theoretical analysis of these methods.

As in [3], we see in further details the two classes of methods inheriting the main features of the CS algorithm, namely *pattern search* (PS) and *mesh adaptive direct search* (MADS).

Pattern Search

Pattern Search [58, 85] is a first generalization of the CS algorithm. At each iteration, starting from the *incumbent* point, that is, the best candidate found by the algorithm so far, the method performs *exploratory* moves following geometric patterns. Such patterns are defined by the directions the search is performed over, which are chosen during the initialization. Using the same step length for each direction, the algorithm is able to produce a sequence of points lying on a *rational lattice*. In modern literature, the lattice is called *mesh*. In the case of the CS algorithm, using the coordinate vectors, the mesh is a grid. Furthermore, when the function is differentiable, one wants to make sure that a descent direction exists among the ones used to define the pattern. In fact, it is possible to ensure such property by using a suitably defined set of $n+1$ directions. Such sets are called *positive bases*. The theory behind positive bases has been studied in the past decades [37, 31, 55, 56]. The scheme of PS can be summarized in the following.

Algorithm 1 Pattern Search

```
given $\mathbf{x}_0 \in \mathbb{R}^n$, $\theta \in (0,1)$, $\alpha_0 > 0$, a positive basis $\mathcal{D} := \{\mathbf{d}_1, \dots, \mathbf{d}_r\}$ with $\mathbf{d}_i \in \mathbb{R}^n$
for $k = 0, 1, \dots$ do
    if $f(\mathbf{x}_k + \alpha_k \mathbf{d}) < f(\mathbf{x}_k)$ for some $\mathbf{d} \in \mathcal{D}$ then
        set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}$ and $\alpha_{k+1} = \alpha_k$
    else
        set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\alpha_{k+1} = \theta \alpha_k$
    end if
end for
```

As we can see, the mesh is not explicitly invoked by the algorithm. Nevertheless, exploiting the mesh is one of the key steps to prove convergence. Additionally, it is reasonable to think that using always the same directions might not be the most efficient way to solve the problems, indeed it would be desirable to use directions that better encompass the local behaviour of the objective function. Having that in mind, and noticing the theoretical importance of the mesh, PS was further generalized.

First, it is possible to generalize the construction of positive bases. Given a matrix $\mathbf{Z} \in \mathbb{Z}^{n \times r}$ which columns form a positive basis for $\mathbb{R}^n$, which is also defined as a *positive spanning matrix*, and an invertible matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$, it is possible to show that the set of columns of the matrix defined as $\mathbf{D} = \mathbf{G}\mathbf{Z}$ is also a positive basis. One can select any finite dimensional matrix $\mathbf{Z}$, so that the number of its columns can be arbitrarily large. The pattern is then defined by the set of columns of $\mathbf{D}$, namely $\mathcal{D}$. Note that it is not necessary to consider the full pattern at each iteration k, but a set $\mathcal{D}_k \subseteq \mathcal{D}$ can be consider, as long as $\mathcal{D}_k$ is also a positive basis. A mesh centered on the current iterate can then be explicitly defined, that is $\mathcal{M}_k = \{\mathbf{x}_k + \delta_k \mathbf{D}\mathbf{y} : \mathbf{y} \in \mathbb{N}^r\} \subset \mathbb{R}^n$. Since the directions in $\mathcal{D}$ do not necessarily share the same norm value, it would not be proper to refer to the parameter δ as the step-size, as we did for α_k, indeed it was first called the *coerciveness* and then standardized into the *mesh-size* parameter. The mesh can be used to define heuristic strategies to find a better incumbent point. As long as the new incumbent lies on the mesh, it does not matter which strategy is used, the algorithm maintains its convergence properties. The latter modifications allow us to introduce the modern class of PS methods, referred to as *generalized pattern search* (GPS) [85]. The structure of GPS is detailed below.

The method is now divided into two main steps: the Search Step, where heuristics are applied and

Algorithm 2 Generalized Pattern Search

```
given initial point $\mathbf{x}_0 \in \mathbb{R}^n$, mesh-size adjustment parameter $\tau \in (0,1)$, initial mesh-size parameter $\delta_0 > 0$, a positive spanning matrix $\mathbf{D} := \mathbf{GZ}$.
for $k = 0, 1, \dots$ do
    Search Step
    if $f(\mathbf{t}) < f(\mathbf{x}_k)$ for some $\mathbf{t} \in \mathcal{S}_k \subseteq \mathcal{M}_k$ then
        set $\mathbf{x}_{k+1} = \mathbf{t}$ and $\delta_{k+1} = \tau^{-1}\delta_k$
    end if
    Poll Step
    Select a positive spanning set $\mathcal{D}_k \subseteq \mathcal{D}$
    if $f(\mathbf{x}_k + \delta_k \mathbf{d}) < f(\mathbf{x}_k)$ for some $\mathbf{d} \in \mathcal{D}_k$ then
        set $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k \mathbf{d}$ and $\delta_{k+1} = \tau^{-1}\delta_k$
    else
        set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\delta_{k+1} = \tau\delta_k$
    end if
end for
```

points are projected to the mesh, identifying a set of trial points $\mathcal{S}_k \subseteq \mathcal{M}_k$; the Poll Step, where exploratory moves along some of the initially defined directions are performed. Note that the Poll Step only utilized directions in $\mathcal{D}$, so that the points generated lie on the mesh by definition. One more difference lies in the possibility for GPS to increase the mesh-size parameter when a better incumbent is found.

Mesh Adaptive Direct Search

The flexibility of choice in GPS makes it a very powerful tool to solve BBO problems. The theory is sound when the objective function is assumed to be differentiable, though, that does not totally solve the challenges of general BBO. Indeed, it is not rare having to deal with nonsmooth functions for such class of problems. Clearly, working with a finite set of directions is not a viable option to reach sound theoretical convergence properties if one does not assume the objective function to be smooth. In order to be able to build an infinite set of directions, GPS is further modified. Surprisingly, the construction of $\mathcal{D}$ does not change, it is still defined as the set of columns of the finite dimensional matrix $\mathbf{D} = \mathbf{GZ}$. The difference lies in the way the directions are generated during the Poll Step. First, the new concept of *frame* is introduced, namely $\mathcal{F}_k = \{\mathbf{x} \in \mathcal{M}_k : \|\mathbf{x}_k - \mathbf{x}\|_\infty \leq \Delta_k b\}$, where $b = \max\{\|\mathbf{d}\|_\infty : \mathbf{d} \in \mathcal{D}\}$, and Δ_k is called the *frame-size* parameter. Using the infinity norm, the frame defines a square centered on the incumbent point wherein all the points lying on the mesh are considered. The set of directions in the Poll Step is now defined as $\mathcal{D}_k^\Delta$ such that the set of points $\mathcal{P}_k = \{\mathbf{x}_k + \delta_k \mathbf{d} : \mathbf{d} \in \mathcal{D}_k^\Delta\}$ is a subset of $\mathcal{F}_k$. Note that the directions in $\mathcal{D}$ now are only defining the geometrical structure of the mesh, and they are not directly affecting the directions in $\mathcal{D}_k^\Delta$ anymore. The idea is that, by letting δ_k go to 0 faster than Δ_k, the set of points in the frame become denser and denser, so that an asymptotically dense set of directions can be built. In order to achieve that, at each iteration k the mesh-size parameter is updated as $\delta_k = \min\{\Delta_k, \Delta_k^2\}$. These modification define the algorithm proposed in [4] and denoted as *Mesh Adaptive Direct Search* (MADS). Note that GPS can be seen as a subclass of MADS where $\delta_k = \Delta_k$.

MADS was developed not only to deal with nonsmooth problems, but also to target constrained ones. Indeed the following problem is considered

$$\min_{\mathbf{x}\in\Omega} f(\mathbf{x}),$$

where $\Omega \subseteq \mathbb{R}^n$ represents the feasible set. The strategy initially proposed to solve such problem is the so-called *extreme barrier*, which consists in considering the following *extended valued barrier function*

$$f_\Omega(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{when } \mathbf{x} \in \Omega \\ \infty & \text{when } \mathbf{x} \notin \Omega \end{cases},$$

and solving the unconstrained problem $\min_x \{f_\Omega(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$. The following scheme describes the main structure of the method.

Algorithm 3 Mesh Adaptive Direct Search

```
given initial point $\mathbf{x}_0 \in \Omega$, frame-size adjustment parameter $\tau \in (0,1)$, initial frame-size parameter $\Delta_0 > 0$, a positive spanning matrix $\mathbf{D} := \mathbf{GZ}$.
for $k = 0, 1, \dots$ do
    Parameter Update
    $\delta_k = \min\{\Delta_k, \Delta_k^2\}$
    Search Step
    if $f_\Omega(\mathbf{t}) < f_\Omega(\mathbf{x}_k)$ for some $\mathbf{t} \in \mathcal{S}_k \subseteq \mathcal{M}_k$ then
        set $\mathbf{x}_{k+1} = \mathbf{t}$ and $\Delta_{k+1} = \tau^{-1}\Delta_k$
    end if
    Poll Step
    Select a set of directions $\mathcal{D}_k^\Delta$ such that $\mathcal{P}_k$ is a subset of $\mathcal{F}_k$
    if $f_\Omega(\mathbf{t}) < f_\Omega(\mathbf{x}_k)$ for some $\mathbf{t} \in \mathcal{P}_k$ then
        set $\mathbf{x}_{k+1} = \mathbf{t}$ and $\delta_{k+1} = \tau^{-1}\delta_k$
    else
        set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\Delta_{k+1} = \tau\Delta_k$
    end if
end for
```

Note that the Search Step is much freer than the Poll Step, since the points generated do not have to lie inside the frame.

1.1.2 Line-search Methods

DFO Line-search arises from a direct modification of the line-search methods based on derivatives. These algorithms generate a sequence of points of the type

$$\mathbf{x}_{k+1} = \mathbf{x} + \alpha_k \mathbf{d}_k.$$

Setting $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ for each k we get the steepest descent. That does not have to be the case in general. Nevertheless, roughly speaking, $\mathbf{d}_k$ has to be a descent direction in order for the algorithm to be globally convergent. Line-search methods are especially focused on determining the right value of the step-size α_k. Well-known conditions to fulfill such purpose are the ones of Armijo or Goldstein, for example. In [38] the authors introduced acceptability criteria for the line-search step-size wherein only objective function values are used. The main idea was to avoid computing the gradient of the objective in intermediate points while computing a suitable length for the step. In particular, the authors consider two of the proposed conditions to be the derivative-free equivalent of Armijo and Goldstein conditions. Since the work did not provide strategies to identify a descent direction without using derivatives, it does not provide a full framework for a derivative-free LS method. Such framework is proposed in [53], where conditions on the sequence of directions are provided in order to guarantee global convergence of the algorithm. In more recent years, LS methods have been proposed using a set of directions over which the algorithm iterates at each iteration k, similarly to the patterns of DS. As we have seen, for classical DS methods new points are accepted when they simply provide a decrease of the objective function, i.e. $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, and it is called *simple decrease*. LS methods do not rely on a mesh for theoretical guarantees, instead they rely on different rules to accept new points. Such rules have the form of

$$f(\mathbf{x}_k + \alpha\mathbf{d}) \leq f(\mathbf{x}) - q(\alpha),$$

where $q : \mathbb{R}_+ \to \mathbb{R}_+$ is known as *sufficient decrease*. The expression of $q(\cdot)$ used the most in literature is $q(\alpha) = \gamma\alpha^2$, with $\gamma > 0$, proposed in [38]. In the same paper, the authors provide a geometrical interpretation of such condition. It enforces a quadratic decrease of the objective function with respect to the step performed. Considering the direction $\mathbf{d}$, a concave parabola is built placing its vertex on the current point $\mathbf{x}_k$. Given an initial step-size α satisfying $f(\mathbf{x}_k + \alpha\mathbf{d}) \leq f(\mathbf{x}) - \gamma\alpha^2$, the step length is allowed to grow as long as the objective function lies below the parabola.

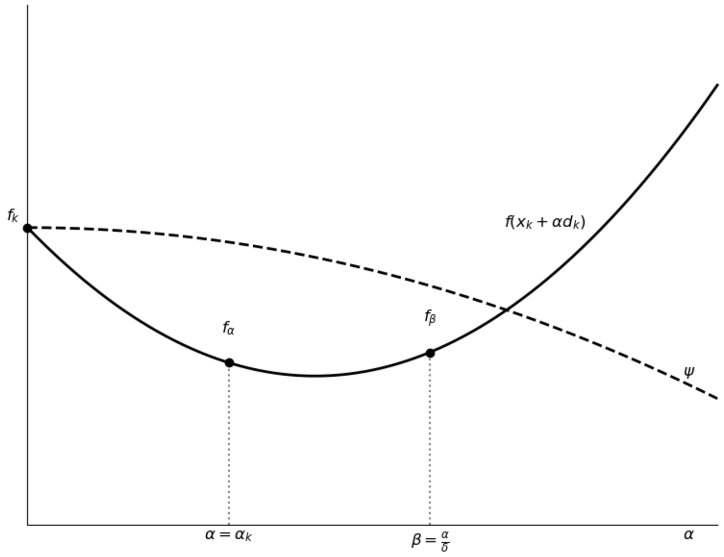

Figure 1.1. Quadratic decrease of the objective function.

Differently from DS, where the iteration k is terminated as soon as a better incumbent is found, LS iterates over all the directions at each iteration k, possibly performing multiple movements. In section 2.3.2 we point out how such feature combined with the behaviour of the line-search provides stronger convergence properties than DS. Another distinguishing feature of LS algorithms is that, considering a set of directions $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_r\}$, different step-sizes α^i are used for the different directions $\mathbf{d}_i$. A formal description of standard LS is detailed below.

Algorithm 4 Line-search

```
given $\mathbf{x}_0 \in \mathbb{R}^n$, $\theta \in (0,1)$, $\delta \in (0,1)$, $\gamma > 0$, $\tilde{\alpha}_0^i > 0$, $i = 1, \ldots, n$, $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_r\}$
for $k = 0, 1, \ldots$ do
    set $\mathbf{y}_k^1 = \mathbf{x}_k$
    for $i = 1, \ldots, r$ do
        compute $\alpha_k^i$ by the DF-Linesearch-Procedure($\mathbf{y}_k^i, \mathbf{d}_i, \gamma, \delta, \tilde{\alpha}_k^i$)
        set $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i + \alpha_k^i \mathbf{d}_k^i$
        if $\alpha_k^i = 0$ then
            set $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$
        else
            set $\tilde{\alpha}_{k+1}^i = \alpha_k$
        end if
    end for
    set $\mathbf{x}_{k+1}$ such that $f(\mathbf{x}_{k+1}) \leq f(\mathbf{y}_k^{r+1})$
end for
```

As we can see, at each iteration k, two kinds of step-sizes appear for each direction $\mathbf{d}_i$:

- α_k^i, the *moving* step-size, which represents the actual movement performed along $\mathbf{d}_i$ at each iteration k, possibly equal to 0;
- $\tilde{\alpha}_k^i$, the *stored* step-size, which represents the initial step-size along $\mathbf{d}_i$ at each iteration k, and, for the iteration $k+1$, it is set equal to α_k^i whenever $\alpha_k^i > 0$ whereas it is reduced by a factor $\theta \in (0,1)$ whenever $\alpha_k^i = 0$.

The moving step-size α_k^i is computed by means of a derivative-free line-search procedure described in the following.

Algorithm 5 DF-Linesearch-Procedure($\mathbf{x}$,$\mathbf{p}$,γ,δ,ν)

```
set $\alpha = \nu$
if $f(\mathbf{x} + \alpha \mathbf{p}) > f(\mathbf{x}) - \gamma \alpha^2$ then
    set $\alpha = 0$ and return $\alpha$
else
    while $f(\mathbf{x} + \frac{\alpha}{\delta}\mathbf{p}) \leq f(\mathbf{x} + \alpha \mathbf{p}) - \gamma \left(\frac{\alpha}{\delta}\right)^2$ do
        set $\alpha = \frac{\alpha}{\delta}$
    end while
end if
return d and $\alpha$
```

Such procedure first makes sure that the objective function of the first trial point lies below the parabola, as described previously. Then, it increases the step-size by repeatedly dividing it by a scalar $\delta \in (0,1)$, until the next point fails to satisfy the sufficient decrease condition, i.e. the procedure stops as soon as a point is found where the objective function lies above the parabola. Such procedure guarantees that at each iteration k and for all $\mathbf{d}_i \in \mathcal{D}$, a step-size $\beta_k^i > 0$ exists such that

$$f(\mathbf{y}_k^i + \beta_k^i \mathbf{d}_i) > f(\mathbf{y}_k^i) - \gamma(\beta_k^i)^2.$$

If the objective function is differentiable and the gradient is Lipschitz continuous, and provided that the set $\mathcal{D}$ positively spans $\mathbb{R}^n$, the latter property allows to determine an upper bound of the norm of the gradient of f for the point $\mathbf{x}_k$ at each iteration k. Further details are provided in 2.3.2. Finally, the line 13 of `Line-search` algorithm, where the incumbent point for the next iteration is chosen, allows for any point wherein the objective function does not increase to be selected. A viable option is of course $\mathbf{y}_k^{r+1}$, but any heuristic strategy can be used to identify a better point, like the Search Step within DS methods.

Note that LS methods are usually proposed using the set of coordinate directions, namely $\mathcal{D} = \{\pm\mathbf{e}_1, \dots \pm \mathbf{e}_n\}$. This is done either to ease the theoretical analysis, or to suitably address problems with bounds on the variables. Note that, in such case, it is preferable to use the same step-size α^i for each coordinate i, i.e. for each pair of directions $\pm\mathbf{e}_i$.

For smooth unconstrained minimization problems, it has been proved LS methods generate a sequence of points $\{\mathbf{x}_k\}$ such that $\lim_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| = 0$, which is stronger than the result for DS wherein only a subsequence of the points generated by the algorithm tends to a point where the norm of the gradient is equal to 0. Suitable modifications of LS have been proposed to provide global convergence properties when the objective function is nonsmooth [42].

1.2 Constrained optimization

In this section we introduce techniques to solve the following nonlinear constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^n} \; & f(\mathbf{x}), \\ \text{s.t.} \;\; & g(\mathbf{x}) \le \mathbf{0}_m, \\ & h(\mathbf{x}) = \mathbf{0}_q, \end{aligned} \tag{CP}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}^q$, and $\mathcal{X}$ is a set defined by linear constraints. We refer to the set of feasible points of (CP) as $\mathcal{F} = \{\mathbf{x} \in \mathcal{X} : g_\ell(\mathbf{x}) \le 0,\ h_j(\mathbf{x}) = 0,\ \ell = 1, \dots, m,\ j = 1, \dots, q\}$. Even though approaches to deal with nonsmooth functions exist, it is out of scope for this work. Thus, we assume all the functions to be continuously differentiable.

Many methods have been proposed to solve constrained problems. Let us propose a summary of the main techniques of our interest:

- *Penalty Methods* (PM): penalty methods modify the objective function by adding a penalty term for violating constraints. The idea is to solve an unconstrained problem where constraint violations are penalized in the objective function, forcing the optimizer to find feasible solutions.
- *Barrier Methods* (BM): starting from a feasible point, they ensure that the solution stays within the feasible region by introducing a barrier function that becomes infinite as the solution approaches the boundary of the feasible region.
- *Filter Methods*: they do not combine the objective function and constraints into a single augmented function. Instead, they treat the objective function and constraints separately and maintain a set of acceptable trade-offs between them.

There exist other approaches to address constraints in optimization, *Augmented Lagrangian* [12], *Sequential Quadratic Programming* [14] or *Projection Methods* [10]. It is not our goal to provide a general understanding of all such methods.
In the following sections we see more insights on PM and BM to understand their basic ideas. In Chapter 3 a technique is proposed that is based on a mix of these two strategies.

1.2.1 Optimality Conditions for Constrained Problems

In order to better understand the previous techniques we first introduce some basic concepts of constrained optimization.
It is important to identify stationary points of Problem (CP). In the case of unconstrained problems it is possible to define these points as those where the directional derivative of the objective function is nonnegative for all $\mathbf{d}$ in $\mathbb{R}^n$, i.e. points $\mathbf{x}$ in $\mathbb{R}^n$ where $\|\nabla f(\mathbf{x})\| = 0$. In the constrained case one has to take into account that some, possibly all, directions might not be feasible, so that points where the norm of the gradient is equal to 0 are not the only candidate solutions for the problem. The following well-known result provide a better characterization of the points of our interest when the set $\mathcal{X}$ is equal to $\mathbb{R}^n$.

Proposition 1.2.1 (Fritz-John theorem)**.** *Let* $\mathbf{x} \in \mathcal{F}$ *be a stationary point of Problem* (CP)*. Then there exists a nonzero vector* $[\lambda_0 \ \lambda_1 \ \dots \ \lambda_m \ \mu_1 \ \dots \ \mu_q] \in \mathbb{R}^{1+m+q}$ *such that*

$$\begin{aligned} \lambda_0 \nabla f(\mathbf{x}) + \sum_{\ell=1}^{m} \lambda_\ell \nabla g_\ell(\mathbf{x}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\mathbf{x}) &= 0, && \\ \lambda_\ell g_\ell(\mathbf{x}) &= 0, && \ell = 1, \dots, m, \\ \lambda_0 \geq 0, \quad \lambda_\ell &\geq 0, && \ell = 1, \dots, m. \end{aligned} \tag{1.1}$$

Conditions (1.1) are useful, but not totally satisfying. Indeed, had we split one equality constraint $h_j(\mathbf{x})$ into two inequalities, i.e. $-h_j(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) \leq 0$, any point $\mathbf{x}$ in $\mathcal{F}$ would satisfy (1.1). One might notice that the presence of λ_0 multiplying the gradient of the objective function allows the constraints to possibly play the main role in complying with the latter conditions. In order to avoid the use of such multiplier, it is necessary for the constraints to satisfy some *regularity conditions*, known in literature as *constraint qualifications*. Many constraint qualifications have been proposed in literature, we make use of the one proposed by Mangasarian and Fromovitz.

Definition 1.2.2 (Mangasarian-Fromovitz constraints qualification [13, Proposition 4.3.18])**.** *A point* $\bar{\mathbf{x}} \in \mathcal{X}$ *is said to satisfy the Mangasarian-Fromovitz constraint qualification (MFCQ) if the following two conditions are met:*

(a) *there does not exist a nonzero vector* $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_q)$ *such that:*

$$\left(\sum_{i=1}^{q} \alpha_i \nabla h_i(\mathbf{x}) \right)^T (\mathbf{x} - \bar{\mathbf{x}}) \geq 0, \qquad \forall \mathbf{x} \in \mathcal{X};$$

(b) *there exists* $\mathbf{x} \in \mathcal{X}$ *such that:*

$$\nabla g_\ell(\mathbf{x})^T (\mathbf{x} - \bar{\mathbf{x}}) < 0 \quad \forall \ell \in \mathcal{I}(\mathbf{x}), \quad \nabla h_j(\mathbf{x})^T (\mathbf{x} - \bar{\mathbf{x}}) = 0 \quad \forall j = 1, ..., q$$

where $\mathcal{I}(\mathbf{x}) = \{i : g_i(\mathbf{x}) = 0\}$*.*

Let us now introduce the Lagrangian function associated with Problem (CP), that is

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T g(\mathbf{x}) + \boldsymbol{\mu}^T h(\mathbf{x}),$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^q$ are the *Lagrangian multipliers* associated with the constraints. The Lagrangian function is an important tool to state optimality conditions for constrained problems. Indeed, the following result has been proved (see, for instance, [13]).

Proposition 1.2.3. *Let* $\mathbf{x}^\star \in \mathcal{F}$ *be a local minimum of problem* (CP) *that satisfies the MFCQ. Then,* $\boldsymbol{\lambda}^\star \in \mathbb{R}^m$, $\boldsymbol{\mu}^\star \in \mathbb{R}^q$ *exist such that*

$$\nabla_x L(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)^T (\mathbf{x} - \mathbf{x}^\star) \geq 0 \qquad \forall \mathbf{x} \in \mathcal{X}, \tag{1.2}$$

$$(\boldsymbol{\lambda}^\star)^T g(\mathbf{x}^\star) = 0, \qquad \boldsymbol{\lambda}^\star \geq 0. \tag{1.3}$$

The latter result can be used to define stationary points for Problem (CP).

Definition 1.2.4 (stationary point)**.** *A point* $\mathbf{x}^\star \in \mathcal{F}$ *is said to be a stationary point for problem* (*CP*) *if vectors* $\boldsymbol{\lambda}^\star \in \mathbb{R}^m$ *and* $\boldsymbol{\mu}^\star \in \mathbb{R}^q$ *exist such that* (1.2) *and* (1.3) *are satisfied.*

1.2.2 Penalty and Barrier methods

As previously stated, PM and BM modify the objective function by adding a term depending on the values of the constraints. It can be formally stated in the following way

$$\begin{aligned} \mathrm{P}M: &\qquad z(\mathbf{x}; \varepsilon) = f(\mathbf{x}) + \frac{1}{\varepsilon} p(\mathbf{x}), \\ \mathrm{B}M: &\qquad z(\mathbf{x}; \varepsilon) = f(\mathbf{x}) + \varepsilon b(\mathbf{x}), \end{aligned}$$

where $\varepsilon \in \mathbb{R}$, $p : \mathbb{R}^n \to \mathbb{R}$ and $b : \mathbb{R}^n \to \mathbb{R}$. The penalty function $p(\cdot)$ represents a penalization which grows along with the violation of the constraints, and it is equal to 0 for all points that are feasible. The barrier function $b(\cdot)$ tends to $+\infty$ as the points approach the boundary of the feasible region, and it is set to $+\infty$ for all points that are infeasible. The parameter ε plays a different role in the two methods. Indeed, in PM as ε decreases the penalization grows, pushing the optimization procedure towards the feasible region, whereas in BM the effect of the barrier is reduced when ε becomes smaller. The two methods work somehow in a complementary way, aligning with the extreme barrier as the parameter ε is driven towards 0. Note the BM works from the interior of the feasible region, thus the barrier cannot be used for equality constraints unless one makes use of slack variables. For the seek of simplicity, in this section we consider Problem (CP) having only inequality constraints, that is

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n} \ & f(\mathbf{x}), \\ \text{s.t.} \ & g(\mathbf{x}) \leq \mathbf{0}_m. \end{aligned} \tag{CPI}$$

The *merit function* $z(\cdot)$ is then used to transform Problem (CPI) into the following one

$$\min_{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n} z(\mathbf{x}; \varepsilon).$$

A sequence of decreasing parameters $\{\varepsilon_k\}$ is built such that $\varepsilon_k \to 0$. Both PM and BM solve a sequence of *subproblems* depending on ε_k, finding for each one an optimal solution $\mathbf{x}(\varepsilon_k)$. The goal is for the sequence of solutions $\{\mathbf{x}(\varepsilon_k)\}$ to tend towards an optimal solution of the original problem.

In order to do so, the merit function must be built so that the gradient of $z(\cdot)$ somehow mimics the behaviour of the gradient of the Lagrangian function. The following are among the most common options

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\ell=1}^{m} \max\{0, g_\ell(\mathbf{x})\}^\nu, \quad \nu > 1, \\ b(\mathbf{x}) &= -\sum_{\ell=1}^{m} \log(-g_\ell(\mathbf{x})). \end{aligned}$$

Note that in such case the domain of $b(\cdot)$ is restricted to $\Omega = \{\mathbf{x} \mid g_\ell(\mathbf{x}) < 0 \text{ for all } \ell = 1, \dots, m\}$. Using the latter definitions and denoting by $z_{\mathtt{PM}}(\cdot)$ and $z_{\mathtt{BM}}(\cdot)$ the merit functions of the penalty and the barrier methods respectively, we get the following expressions for their gradients

$$\begin{aligned} \nabla z_{\mathtt{PM}}(\mathbf{x}; \varepsilon_k) &= \nabla f(\mathbf{x}) + \sum_{\ell=1}^{m} \frac{\nu}{\varepsilon_k} \max\{0, g_\ell(\mathbf{x})\}^{\nu-1} \nabla g_\ell(\mathbf{x}), \\ \nabla z_{\mathtt{BM}}(\mathbf{x}; \varepsilon_k) &= \nabla f(\mathbf{x}) + \sum_{\ell=1}^{m} \frac{\varepsilon_k}{-g_\ell(\mathbf{x})} \nabla g_\ell(\mathbf{x}). \end{aligned}$$

It is possible to observe that by defining $\lambda_\ell(\varepsilon_k) = \dfrac{\nu}{\varepsilon_k} \max\{0, g_\ell(\mathbf{x}(\varepsilon_k))\}^{\nu-1}$ for PM and $\lambda_\ell(\varepsilon_k) = \dfrac{\varepsilon_k}{-g_\ell(\mathbf{x}(\varepsilon_k))}$, recalling that $\mathbf{x}(\varepsilon_k)$ is an optimum solution for the subproblem related to ε_k, we get

$$\left(\nabla f(\mathbf{x}(\varepsilon_k)) + \sum_{\ell=1}^{m} \lambda_\ell(\varepsilon_k) \nabla g_\ell(\mathbf{x}(\varepsilon_k))\right)^\top (\mathbf{x} - \mathbf{x}(\varepsilon_k)) =$$

$$\nabla_x L(\mathbf{x}(\varepsilon_k), \boldsymbol{\lambda}(\varepsilon_k))^\top (\mathbf{x} - \mathbf{x}(\varepsilon_k)) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}.$$

Thus, all points of the sequence $\{\mathbf{x}(\varepsilon_k)\}$ satisfy condition (1.2). Unfortunately, one can not generally ensure condition (1.3) to be satisfied by $\mathbf{x}(\varepsilon_k)$ for any value of ε_k. One can either prove that such condition is met by the limit point of the sequence $\{\mathbf{x}(\varepsilon_k)\}$ or assume so. In both cases, a major issue has to be overcome, that is, the sequences $\{\lambda_\ell(\mathbf{x}(\varepsilon_k))\}$ might be unbounded. The matter is that Proposition 1.2.3 requires $\boldsymbol{\lambda}$ to be a real vector, so that the limit of such sequences must be finite. Further assumptions on the problem or on the limit point of the sequence $\{\mathbf{x}(\varepsilon_k)\}$ has to be made in order to ensure the sequences of multipliers to be bounded. Furthermore, for real problems it is nonviable to find an exact solution, so that for numerical implementations of such methods some level of approximation has to be allowed when solving any subproblem.

The mentioned difficulties are all addressed in Chapter 3, where a strategy that mixes the penalty and the barrier approaches is proposed within two different DFO frameworks.

Chapter 2

Worst-case complexity bounds for Line-search DFO methods [*]

The motivation to explore worst-case complexity bounds for line-search methods in derivative-free optimization (DFO) arose from a fundamental observation: line-search methods have shown better convergence properties compared to direct search methods, which are more common in derivative-free contexts. This observation prompted an investigation into whether the inherently stronger convergence characteristics of line-search algorithms could yield more favorable complexity bounds than those already established for direct search methods. Initially, the project aimed to leverage these stronger properties to go beyond the worst-case complexity bounds typically derived for direct search algorithms, hoping to demonstrate that line-search methods could achieve improved iteration efficiency and potentially uncover further advantages.

In particular, while bounding the number of iterations required to drive the gradient norm below a specified tolerance ε for line-search methods followed an approach similar to direct search methods, establishing a bound on the number of function evaluations proved more intricate. The line-search approach inherently allows the step-size to expand multiple times along a given direction, introducing variability in the number of function evaluations required at each iteration. Consequently, in our initial work, detailed in Section 2.3, we conducted a detailed analysis of the line-search method's behavior to identify any necessary conditions to at least match the complexity bounds known for direct search methods, while also exploring the potential for improved bounds or additional theoretical properties unique to line-search.

The results were encouraging. Not only did we achieve bounds comparable to those for direct search methods, but we also reached further by providing a bound on the total number of function evaluations for which the norm of the gradient remains above the tolerance ε, an accomplishment that, to our knowledge, is novel within the DFO context.

This work subsequently led us to consider the expansion procedures characteristic of line-search methods as a potential advantage when dealing with bound-constrained optimization problems. The hypothesis was that the extrapolation capability inherent in line-search methods could facilitate

---

[*] The work presented in this chapter has been developed in two papers. The first, *"Worst Case Complexity Bounds for Linesearch-Type Derivative-Free Algorithms"* [20], co-authored with Prof. Giampaolo Liuzzi ("Sapienza" University of Rome), Prof. Stefano Lucidi ("Sapienza" University of Rome), Prof. Morteza Kimiaei (University of Vienna). The second, *"Complexity results and active-set identification of a derivative-free method for bound-constrained problems"* [18], co-authored with Prof. Andrea Cristofari (University of Rome "Tor Vergata"), Prof. Giampaolo Liuzzi (–), Prof. Stefano Lucidi (–).

faster convergence toward bounds, especially compared to other DFO methods. Inspired by this potential, we extended our investigation to derive worst-case complexity bounds for bound-constrained problems akin to those established in the unconstrained case. A key aim here was to demonstrate the algorithm's ability to identify the active set of the final solution—those variables resting on the boundary constraints—in a finite number of iterations. This bound-constrained analysis and its implications are detailed in Section 2.4, completing a comprehensive study of line-search methods in both unconstrained and bound-constrained settings within the DFO framework.

Through this exploration, Chapter 2 establishes line-search methods as a robust and theoretically grounded approach to DFO.

2.1 Introduction

In this section we provide an introduction to the topics of this chapter. In particular we first explore some literature related to the worst-complexity bounds derived for unconstrained problems in the context of DFO and we underline our contribution to this topic. Then, we describe the main known results for complexity bounds related to bound (and linear) constrained optimization and we deepen into the contributions of our work.

Following the introductory section, there are four more: Section 2.2 introduces the algorithm that constitutes the basis for both unconstrained and bound-constrained problems, that is, the `LAM` algorithm; in Section 2.3 and Section 2.4 we analyze the properties of `LAM` to unconstrained and bound-constrained problems respectively; in Section 2.5 we derive some final remarks on the work presented.

2.1.1 Unconstrained case: literature and contribution

In the unconstrained settings, over the past decade, the analysis of worst case complexity for optimization algorithms has gained more and more interest and attracted many researchers [25, 2]. Specifically for derivative-free algorithms, in [86, 40] worst case complexity bounds have been derived for direct search methods using sufficient decrease in f. In particular, it has been proved that direct search methods (based on a search step and a poll step and using sufficient decrease in the objective function to accept new points) require at most $\mathcal{O}(n\epsilon^{-2})$ iterations and $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations to find a point $\mathbf{x}_k$ such that $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$.

In [24] an adaptive cubic regularization algorithm has been proposed which is based on gradient estimation via finite differences. The algorithm has a worst case complexity of $\mathcal{O}(n^2\epsilon^{-3/2})$ function evaluations, which is better than the complexity obtained for direct search methods. Additionally, a method based on quadratic regularization with finite-difference gradient approximations has been recently proposed [50], achieving a worst-case complexity bound of $\mathcal{O}(n\epsilon^{-2})$ function evaluations.

In [48], the complexity of a smoothing technique for the optimization of nonsmooth functions has been studied. It has been shown that the smoothing algorithm has a worst case complexity of $\mathcal{O}(\epsilon^{-3})$ to achieve ϵ-stationarity.

Prior to this thesis, analogous results for line-search based DFO algorithms (see e.g. [75, 42]) have not yet been established. These algorithms typically have stronger asymptotic convergence properties which are tied to the use of more complex extrapolation techniques.

For unconstrained problems, we contribute to the DFO literature by proposing a framework of DFO algorithms based on a line-search-type extrapolation technique, which exploits sufficient decrease, and by proving that such algorithms share the same worst-case complexity bounds of direct search methods. The results heavily depend on the ability of the algorithm to produce sufficient decrease of an auxiliary function which encompasses the overall behaviour of the method, regardless of the iteration being successful or unsuccessful. Furthermore, thanks to the extrapolation technique, it is possible to prove that the number of iterations (in the worst case) wherein $\|\nabla f(\mathbf{x}_k)\| \geq \epsilon$ is of the order of ϵ^{-2}. The latter property considerably enriches the worst-case analysis of DFO algorithms and, to the best of our knowledge, is new in this context. The property characterizes the behaviour of the DFO algorithm better than the usual complexity results. Indeed, typical complexity results provide a bound for the number of iterations required to drive the norm of the gradient below a prefixed tolerance for the first time. When the method is running, the norm of the gradient might well rise back above the tolerance again. The property we propose bounds the total number of iterations wherein the norm of the gradient lies above a specified tolerance by a constant that depends on ϵ^{-2}.

2.1.2 Bound-constrained case: literature and contribution

Several derivative-free methods have been proposed to deal with the presence of bounds or linear constraints. In particular, we can distinguish among model-based methods [27, 30, 52, 54, 59, 83], where the objective function is sampled in a neighborhood of the current iterate to build an appropriate model to be minimized, direct search methods [4, 51, 60, 61, 65], where the objective function is sampled in a neighborhood of the current iterate in order to find a descent direction, and line-search methods [74, 76], where directions are explored by allowing the step-size to dynamically expand and contract.

For model-based and direct search methods applied to problems with linear constraints (thus including bound constraints), a worst-case analysis can be found in [51, 59], providing upper bounds on the maximum number of iterations and function evaluations needed to drive a criticality measure below a prespecified threshold. In particular, in [51], it is shown that at most $\mathcal{O}(n\epsilon^{-2})$ iterations and $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations are required, for a (deterministic) direct search method, to produce the first point with a criticality measure below $\epsilon > 0$, matching the same complexity for the unconstrained case [40, 86]. In [59], similar bounds of $\mathcal{O}(k_D^2\epsilon^{-2})$ iterations and $\mathcal{O}(nk_D^2\epsilon^{-2})$ function evaluations are obtained, matching the same complexity for the unconstrained case [47], with k_D being a problem dimension-dependent constant which define a fully linear model.

Here we contribute to the DFO literature by extending the framework proposed in [20] for unconstrained problems to bound-constrained ones.

We provide a worst-case analysis for the proposed method, which yields to the same bounds for direct search [51], that is, $\mathcal{O}(n\epsilon^{-2})$ iterations and $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations to produce the first point with a criticality measure below $\epsilon > 0$. Additionally, as in the unconstrained case, we are able to bound the *total* number of iterations wherein the criticality measure is above ϵ, thus reaching beyond the complexity results for direct search methods given in [51].

Furthermore, we are able to prove identification of the active constraints in a finite number of iterations for the proposed line-search method. Such a property is usually desirable for an optimization algorithm due to, among other things, the possibility of saving function evaluations if one recognizes the surface where the optimal solution lies. In several applications, we might be interested only in the identification of the surface containing an optimal solution (or its support). In the literature, finite active-set identification was established for smooth optimization algorithms and proximal methods (see, e.g., [11, 15, 22, 57, 87]), also providing complexity bounds in some cases [16, 32, 81]. For

DFO, parameter-dependent estimates were used in [52, 68, 76], allowing for finite identification of active constraints if certain conditions hold. Moreover, finite identification results have been shown in [33] for a method using an inner approximation approach to minimize a function over the convex hull of a given set of vectors, meaning that, in a finite number of iterations, the algorithm is able to identify, under appropriate assumptions, the vectors with zero weight in the convex combination representing the final solution. Here, we prove that the proposed algorithm correctly identifies the active constraints satisfying the strict complementarity condition in a finite number of iterations, without using any parameter-dependent estimate. Namely, this feature is a general property of the method.

2.2 The problem and the LAM algorithm

We consider the following minimization problem

$$\min_{\mathbf{x}\in\Omega\subseteq\mathbb{R}^n} f(\mathbf{x}), \tag{P1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is a black-box function which is known by means of an oracle that only outputs function values. Hence, derivatives of f can neither be approximated nor computed explicitly. The feasible set Ω is equal to $\mathbb{R}^n$ in the unconstrained case, while for bound-constrained problems it is defined as $\Omega = \{\mathbf{x}\in\mathbb{R}^n : \; l_i \le x_i \le u_i, \; i = 1,\dots,n\}$, where $l_i, u_i \in \mathbb{R}\cup\{\pm\infty\}$, $l_i < u_i$, $i = 1,\dots,n$. For the sake of clarity we use the notation $[\mathbf{l},\mathbf{u}]$ to refer to the feasible set Ω in such case.

Throughout the present chapter, for both unconstrained and bound-constrained problems, the following assumption is considered satisfied, even where not explicitly invoked.

Assumption 2.1. *The objective function $f:\mathbb{R}^n\to\mathbb{R}$ is continuously differentiable with a Lipschitz continuous gradient ∇f with constant $L>0$ over Ω, i.e.,*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\| \quad \forall \mathbf{x},\mathbf{y}\in\Omega.$$

Moreover, f is bounded from below over Ω, i.e., a constant $f_{\min}\in\mathbb{R}$ exists such that

$$f_{\min} \le f(\mathbf{x}) \quad \forall \mathbf{x}\in\Omega.$$

To solve problem P1, we propose a DFO algorithm based on an extrapolation technique, and we refer to it as `Linesearch Algorithm Model` (`LAM`). The main structure of `LAM` is the same for both unconstrained and bound-constrained problems, the difference between the two cases lies in the `DF-Linesearch`, the procedure where the extrapolation technique is performed, wherein the bounds are explicitly treated in their presence. The details of the `DF-Linesearch` procedures are provided in the following sections, and they are analyzed differently for the two cases we are considering.

As we can see, at each iteration k, `LAM` performs an exploration of the space around the current iterate $\mathbf{x}_k$ using the coordinate directions and producing the points $\mathbf{y}_k^i$, $i = 1,\dots,n+1$ (note that $\mathbf{y}_k^1 = \mathbf{x}_k$).

More in particular, points $\mathbf{y}_k^i$, for $i = 2,\dots,n+1$, are computed by means of a suitable derivative-free line-search, namely the `DF-Linesearch` procedure. The line-search is invoked by passing a tentative step-size, i.e.

$$\nu_k^i = \max\{\tilde{\alpha}_k^i, c\Delta_k\},$$

Algorithm 6 Linesearch Algorithm Model: LAM

```
given $\mathbf{x}_0 \in \Omega$, $\theta \in (0,1)$, $\delta \in (0,1)$, $\gamma > 0$, $c \in (0,1]$, $\tilde{\alpha}_0^i > 0$, $i = 1,\dots,n$
for $k = 0,1,\dots$ do
    set $\Delta_k = \max_{i=1,\dots,n}\{\tilde{\alpha}_k^i\}$
    set $\mathbf{y}_k^1 = \mathbf{x}_k$
    for $i = 1,\dots,n$ do
        set $\nu_k^i = \max\{\tilde{\alpha}_k^i, c\Delta_k\}$
        compute $\mathbf{d}_k^i$ and $\alpha_k^i$ by the DF-Linesearch($\mathbf{y}_k^i, i, \gamma, \delta, \nu_k^i$)
        set $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i + \alpha_k^i \mathbf{d}_k^i$
    end for
    set $\mathbf{x}_{k+1} = \mathbf{y}_k^{n+1}$
    if $\mathbf{x}_{k+1} \neq \mathbf{x}_k$ then
        set $\tilde{\alpha}_{k+1}^i = \begin{cases} \alpha_k^i & \text{if } \alpha_k^i > 0 \\ \nu_k^i & \text{otherwise} \end{cases} \qquad i = 1,\dots,n$
    else
        set $\tilde{\alpha}_{k+1}^i = \theta\nu_k^i$, $i = 1,\dots,n$
    end if
end for
```

where $\Delta_k = \max_{j=1,\dots,n}\{\tilde{\alpha}_k^j\}$. It is worth noting that the initial step-size ν_k^i is quite unusual for linesearch-type derivative-free algorithms. In this work, such initialization is of paramount importance since it allows us to prove that within all the iterations the method achieves sufficient decrease of an auxiliary function that is used to derive the worst-case complexity bounds.

During the extrapolation phase, an actual step-size, i.e. α_k^i, is produced, which can either be 0 or greater or equal than ν_k. We refer to k as a *successful iteration* if $\mathbf{x}_{k+1} \neq \mathbf{x}_k$, indicating that at least one positive step-size α_k^i, $i = 1,\dots,n$, has been computed. We refer to k as an *unsuccessful iteration* if $\mathbf{x}_{k+1} = \mathbf{x}_k$, that is, if $\alpha_k^i = 0$ for all $i = 1,\dots,n$. Depending on whether the iteration k is successful or unsuccessful, we use specific rules to update the tentative step-sizes for the next iteration $k+1$. In further detail, for a successful iteration k, each $\tilde{\alpha}_{k+1}^i$ is set to α_k^i if the latter is positive, whereas $\tilde{\alpha}_{k+1}^i$ is set to ν_k^i otherwise. For an unsuccessful iteration k, each $\tilde{\alpha}_{k+1}^i$ is set to $\theta\nu_k^i$, with $\theta \in (0,1)$.

2.3 Unconstrained problems

In this section we see the insights of the framework proposed to solve P1 in the case where $\Omega = \mathbb{R}^n$. We start by introducing the line-search within the algorithm in such case and describe its main features, afterwards we prove convergence properties of `LAM`, and finally we prove the worst-case complexity bounds in terms of iterations and function evaluations.

2.3.1 Unconstrained DF-Linesearch

In the following section we introduce the unconstrained extrapolation technique.

As we can see, The `DF-Linesearch` employs a sufficient decrease criterion, which is evaluated

Algorithm 7 Unconstrained DF-Linesearch($\mathbf{x}$,i,γ,δ,ν)

```
1:  set $\alpha = \nu$ and $\mathbf{d} = \mathbf{e}^i$
2:  if $f(\mathbf{x} + \alpha\mathbf{d}) \leq f(\mathbf{x}) - \gamma\alpha^2$ then
3:      go to line 9
4:  end if
5:  if $f(\mathbf{x} - \alpha\mathbf{d}) \leq f(\mathbf{x}) - \gamma\alpha^2$ then
6:      set $\mathbf{d} = -\mathbf{e}^i$ and go to line 9
7:  end if
8:  set $\alpha = 0$ and return d and $\alpha$
9:  while $f(\mathbf{x} + \frac{\alpha}{\delta}\mathbf{d}) \leq f(\mathbf{x} + \alpha\mathbf{d}) - \gamma\left((\frac{1}{\delta} - 1)\alpha\right)^2$ do
10:     set $\alpha = \frac{\alpha}{\delta}$
11: end while
12: return d and $\alpha$
```

between successive points, i.e.

$$f(\mathbf{x} + \frac{\alpha}{\delta}\mathbf{d}) \leq f(\mathbf{x} + \alpha\mathbf{d}) - \gamma\left((\frac{1}{\delta} - 1)\alpha\right)^2.$$

Regarding the line-search (expansion) procedure, its primary distinction from other line-search techniques (see e.g. [74, 75]) lies in its continual expansion of the step while maintaining adequate decrease between consecutive points. We note that other derivative-free line-search approaches employ the following sufficient decrease criterion

$$f\left(\mathbf{x} + \frac{\alpha}{\delta}\mathbf{d}\right) \leq f(\mathbf{x}) - \gamma\left(\frac{\alpha}{\delta}\right)^2.$$

The above formula uses as reference point the initial point $\mathbf{x}$, whereas the criterion within the method we are proposing uses as reference point $\mathbf{x} + \alpha\mathbf{d}$.

The adopted sufficient decrease criterion allows us to obtain the complexity bound on the number of function evaluations in a more straightforward way than the usual criterion. Needless to say, both criteria lead to the same asymptotic global convergence properties.

To facilitate the analysis of the convergence properties of the algorithm and to deepen further into the overall behaviour of `LAM`, we report and prove the following property, which connects the value of the maximum of the step-sizes between consecutive iterations.

Lemma 2.3.1. *Let $\{\Delta_k\}$ be the sequence of maximum tentative step-sizes produced by `LAM` algorithm using the `Unconstrained DF-Linesearch`. Then, for all k we have*

$$\Delta_{k+1} \begin{cases} \geq \Delta_k & \text{if } \mathbf{x}_{k+1} \neq \mathbf{x}_k \\ = \theta\Delta_k & \text{if } \mathbf{x}_{k+1} = \mathbf{x}_k. \end{cases}$$

Therefore, $\Delta_k \leq \Delta_{k+1}/\theta$ for all k.

Proof. First, let us consider the case where $\mathbf{x}_{k+1} \neq \mathbf{x}_k$ (i.e., k is a successful iteration). In this case, the algorithm sets $\tilde{\alpha}^i_{k+1} = \alpha^i_k$ if $\alpha^i_k > 0$ and $\tilde{\alpha}^i_{k+1} = \nu^i_k$ if $\alpha^i_k = 0$. Since, from the line-search procedure, $\alpha^i_k \geq \nu^i_k$ for all $i = 1, \dots, n$, we can write

$$\tilde{\alpha}^i_{k+1} \geq \nu^i_k \geq \tilde{\alpha}^i_k, \quad i = 1, \dots, n,$$

where the last inequality follows from the definition of ν_k^i. Then, using the definition of Δ_k, we get $\Delta_{k+1} \geq \Delta_k$.

Now, let us consider the case where $\mathbf{x}_{k+1} = \mathbf{x}_k$ (i.e., k is an unsuccessful iteration). In this case, the algorithm sets $\tilde{\alpha}_{k+1}^i = \theta \nu_k^i = \theta \max\{\tilde{\alpha}_k^i, c\Delta_k\}$. Namely, for all $i = 1, \dots, n$, we have that

$$\tilde{\alpha}_{k+1}^i = \begin{cases} \theta \tilde{\alpha}_k^i & \text{if } \tilde{\alpha}_k^i \geq c\Delta_k, \\ \theta c \Delta_k & \text{if } \tilde{\alpha}_k^i < c\Delta_k. \end{cases} \tag{2.1}$$

From the definition of Δ_k and the fact that $c \in (0, 1]$, it follows that $\tilde{\alpha}_{k+1}^i \leq \theta \Delta_k$ for all $i = 1, \dots, n$, implying that

$$\Delta_{k+1} = \max_{i=1,\dots,n} \tilde{\alpha}_{k+1}^i \leq \theta \Delta_k. \tag{2.2}$$

Now, let $\bar{\imath} \in \{1, \dots, n\}$ be such that $\tilde{\alpha}_k^{\bar{\imath}} = \Delta_k$. Since $c \in (0, 1]$, we have $\tilde{\alpha}_k^{\bar{\imath}} \geq c\Delta_k$ and then, recalling (2.1), we get $\tilde{\alpha}_{k+1}^{\bar{\imath}} = \theta \tilde{\alpha}_k^{\bar{\imath}} = \theta \Delta_k$. It follows from (2.2) that

$$\Delta_{k+1} = \tilde{\alpha}_{k+1}^{\bar{\imath}} = \theta \Delta_k,$$

thus concluding the proof. □

2.3.2 Asymptotic convergence analysis for LAM

First of all, we derive an upper bound on the norm of $\nabla f(\mathbf{x}_k)$ at each iteration k.

Proposition 2.3.2. *Suppose that Assumption 2.1 holds. Let $\{\mathbf{x}_k\}$ be the sequence produced by* `LAM` *using* `Unconstrained DF-Linesearch`*. Then, for each k such that $\mathbf{x}_{k+1} \neq \mathbf{x}_k$*

$$\|\nabla f(\mathbf{x}_k)\| \leq \sqrt{n} \left(\frac{\gamma + L(\sqrt{n} + 1)}{\delta} \right) \Delta_{k+1}, \tag{2.3}$$

whereas, for each k such that $\mathbf{x}_{k+1} = \mathbf{x}_k$

$$\|\nabla f(\mathbf{x}_k)\| \leq \sqrt{n} \frac{\gamma + L}{\theta} \Delta_{k+1}, \tag{2.4}$$

where the constants γ, δ, θ are those defined in the `LAM` *algorithm and the* `Unconstrained DF-Linesearch` *procedure.*

Proof. For each iteration k such that $\mathbf{x}_{k+1} \neq \mathbf{x}_k$ and every index $i = 1, \dots, n$, one of two cases can occur:

Case (i), $\alpha_k^i = 0$. By $\alpha_k^i = 0$, and $\tilde{\alpha}_{k+1}^i = \nu_k^i$, we have:

$$f(\mathbf{y}_k^i + \nu_k^i \mathbf{e}_i) > f(\mathbf{y}_k^i) - \gamma(\nu_k^i)^2,$$
$$f(\mathbf{y}_k^i - \nu_k^i \mathbf{e}_i) > f(\mathbf{y}_k^i) - \gamma(\nu_k^i)^2.$$

Then we get from the Mean Value Theorem

$$\nabla f(\mathbf{u}_k^i)^T \mathbf{e}_i > -\gamma \nu_k^i, \tag{2.5}$$
$$\nabla f(\mathbf{v}_k^i)^T \mathbf{e}_i < \gamma \nu_k^i, \tag{2.6}$$

where $\mathbf{u}_k^i = \mathbf{y}_k^i + \lambda_k^i \nu_k^i \mathbf{e}_i$ and $\mathbf{v}_k^i = \mathbf{y}_k^i - \mu_k^i \nu_k^i \mathbf{e}_i$ with $\lambda_k^i, \mu_k^i \in (0,1)$. From (2.5) and (2.6) and the Lipschitz continuity of ∇f, we have that

$$\begin{aligned}\nabla f(\mathbf{x}_k)^T \mathbf{e}_i &> -\gamma \nu_k^i - L\|\mathbf{x}_k - \mathbf{u}_k^i\| > -\gamma \nu_k^i - L\|\mathbf{x}_k - \mathbf{y}_k^i\| - L\nu_k^i,\\ \nabla f(\mathbf{x}_k)^T \mathbf{e}_i &< \quad \gamma \nu_k^i + L\|\mathbf{x}_k - \mathbf{v}_k^i\| < \quad \gamma \nu_k^i + L\|\mathbf{x}_k - \mathbf{y}_k^i\| + L\nu_k^i.\end{aligned}$$

Hence

$$\begin{aligned}|\nabla f(\mathbf{x}_k)^T \mathbf{e}_i| &< (\gamma + L)\nu_k^i + L\|\mathbf{x}_k - \mathbf{y}_k^i\| \le (\gamma + L)\nu_k^i + L\sqrt{n}\Delta_{k+1}\\ &= (\gamma + L)\ \tilde{\alpha}_{k+1}^i + L\sqrt{n}\Delta_{k+1}\\ &\le (\gamma + L)\Delta_{k+1} + L\sqrt{n}\Delta_{k+1},\end{aligned}$$

so that

$$|\nabla f(\mathbf{x}_k)^T \mathbf{e}_i| \le \Big(\gamma + L(\sqrt{n} + 1)\Big)\Delta_{k+1}.$$

Case (ii). From $\alpha_k^i = \alpha$, and $\tilde{\alpha}_{k+1}^i = \alpha \ge \nu_k^i$, it results either

$$\begin{aligned}f\left(\mathbf{y}_k^i + \frac{\tilde{\alpha}_{k+1}^i}{\delta}\mathbf{e}_i\right) &> f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - \gamma\left(\frac{1}{\delta} - 1\right)^2 (\tilde{\alpha}_{k+1}^i)^2,\\ f(\mathbf{y}_k^i + \delta\tilde{\alpha}_{k+1}^i \mathbf{e}_i) &\ge f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) + \gamma(1-\delta)^2(\tilde{\alpha}_{k+1}^i)^2\end{aligned}$$

or

$$\begin{aligned}f\left(\mathbf{y}_k^i - \frac{\tilde{\alpha}_{k+1}^i}{\delta}\mathbf{e}_i\right) &> f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - \gamma\left(\frac{1}{\delta} - 1\right)^2 (\tilde{\alpha}_{k+1}^i)^2,\\ f(\mathbf{y}_k^i - \delta\tilde{\alpha}_{k+1}^i \mathbf{e}_i) &\ge f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) + \gamma(1-\delta)^2(\tilde{\alpha}_{k+1}^i)^2.\end{aligned}$$

Then, we get,

$$\nabla f(\bar{\mathbf{u}}_k^i)^T \mathbf{e}_i > -\gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i, \quad -\nabla f(\hat{\mathbf{u}}_k^i)^T \mathbf{e}_i \ge \gamma(1-\delta)\tilde{\alpha}_{k+1}^i, \tag{2.7}$$

or

$$\nabla f(\bar{\mathbf{v}}_k^i)^T \mathbf{e}_i < \gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i, \quad -\nabla f(\hat{\mathbf{v}}_k^i)^T \mathbf{e}_i \le -\gamma(1-\delta)\tilde{\alpha}_{k+1}^i, \tag{2.8}$$

where $\bar{\mathbf{u}}_k^i = \mathbf{y}_k^i + \bar{\lambda}_k^i\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i \mathbf{e}_i$, $\hat{\mathbf{u}}_k^i = \mathbf{y}_k^i + \hat{\lambda}_k^i(1-\delta)\tilde{\alpha}_{k+1}^i \mathbf{e}_i$, $\bar{\mathbf{v}}_k^i = \mathbf{y}_k^i - \bar{\mu}_k^i\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i \mathbf{e}_i$, and $\hat{\mathbf{v}}_k^i = y_k^i - \hat{\mu}_k^i(1-\delta)\tilde{\alpha}_{k+1}^i \mathbf{e}_i$, with $\bar{\lambda}_k^i, \hat{\lambda}_k^i, \bar{\mu}_k^i, \hat{\mu}_k^i \in (0,1)$.

When (2.7) holds, from $\nabla f(\bar{\mathbf{u}}_k^i)^T \mathbf{e}_i > -\gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i$ we can write

$$[\nabla f(\bar{\mathbf{u}}_k^i) - \nabla f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)]^T \mathbf{e}_i > -\gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i,$$

so that we obtain

$$\begin{aligned}\nabla f(\mathbf{x}_k)^T e_i &> -\gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i - L\|\mathbf{x}_k - \bar{\mathbf{u}}_k^i\|\\ &> -\gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i - L\|\mathbf{x}_k - \mathbf{y}_k^i\| - L\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i.\end{aligned} \tag{2.9}$$

From $\nabla f(\hat{\mathbf{u}}_k^i)^T \mathbf{e}_i \leq -\gamma(1-\delta)\tilde{\alpha}_{k+1}^i$ in (2.7), we can write

$$[\nabla f(\hat{\mathbf{u}}_k^i) - \nabla f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)]^T \mathbf{e}_i \leq -\gamma(1-\delta)\tilde{\alpha}_{k+1}^i,$$

so that, in this case, we obtain

$$\begin{aligned}\nabla f(\mathbf{x}_k)^T \mathbf{e}_i &\leq -\gamma(1-\delta)\tilde{\alpha}_{k+1}^i + L\|\mathbf{x}_k - \hat{\mathbf{u}}_k^i\| \\ &\leq \gamma\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i + L\|\mathbf{x}_k - \mathbf{y}_k^i\| + L\left(\frac{1-\delta}{\delta}\right)\tilde{\alpha}_{k+1}^i.\end{aligned} \tag{2.10}$$

Now, considering (2.9) and (2.10), we get

$$|\nabla f(\mathbf{x}_k)^T \mathbf{e}_i| \leq \left(\frac{\gamma + L(\sqrt{n}+1)}{\delta}\right)\Delta_{k+1}. \tag{2.11}$$

The same bound can be obtained when (2.8) holds. Thus, finally, we obtain

$$\|\nabla f(\mathbf{x}_k)\| \leq \sqrt{n}\left(\frac{\gamma + L(\sqrt{n}+1)}{\delta}\right)\Delta_{k+1}.$$

On the other hand, for each iteration k such that $\mathbf{x}_{k+1} = \mathbf{x}_k$, i.e. $\mathbf{y}_k^i = \mathbf{x}_k$ for all $i = 1, \dots, n+1$, we have for every index $i = 1, \dots, n$

$$\begin{aligned} f\left(\mathbf{x}_k + \frac{\tilde{\alpha}_{k+1}^i}{\theta}\mathbf{e}_i\right) &> f(\mathbf{x}_k) - \gamma\left(\frac{\tilde{\alpha}_{k+1}^i}{\theta}\right)^2, \\ f\left(\mathbf{x}_k - \frac{\tilde{\alpha}_{k+1}^i}{\theta}\mathbf{e}_i\right) &> f(\mathbf{x}_k) - \gamma\left(\frac{\tilde{\alpha}_{k+1}^i}{\theta}\right)^2. \end{aligned}$$

Then we get from the Mean Value Theorem

$$\nabla f(\mathbf{u}_k^i)^T \mathbf{e}_i > -\gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta}, \tag{2.12}$$

$$\nabla f(\mathbf{v}_k^i)^T \mathbf{e}_i < \gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta}, \tag{2.13}$$

where $\mathbf{u}_k^i = \mathbf{x}_k + \lambda_k^i \frac{\tilde{\alpha}_{k+1}^i}{\theta}\mathbf{e}_i$ and $\mathbf{v}_k^i = \mathbf{x}_k - \mu_k^i \frac{\tilde{\alpha}_{k+1}^i}{\theta}\mathbf{e}_i$ with $\lambda_k^i, \mu_k^i \in (0,1)$. From (2.12) and (2.13) and the Lipschitz continuity of ∇f, we have that

$$\begin{aligned} \nabla f(\mathbf{x}_k)^T \mathbf{e}_i &> -\gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta} - L\|\mathbf{x}_k - \mathbf{u}_k^i\| > -\gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta} - L\frac{\tilde{\alpha}_{k+1}^i}{\theta}, \\ \nabla f(\mathbf{x}_k)^T \mathbf{e}_i &< \quad \gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta} + L\|\mathbf{x}_k - \mathbf{v}_k^i\| < \quad \gamma\frac{\tilde{\alpha}_{k+1}^i}{\theta} + L\frac{\tilde{\alpha}_{k+1}^i}{\theta}. \end{aligned}$$

Hence

$$|\nabla f(\mathbf{x}_k)^T \mathbf{e}_i| < (\gamma + L)\frac{\tilde{\alpha}_{k+1}^i}{\theta},$$

so that we can write

$$\|\nabla f(\mathbf{x}_k)\| \leq \sqrt{n}\frac{\gamma + L}{\theta}\Delta_{k+1},$$

concluding the proof. □

Drawing inspiration from [62, 26, 5], we now introduce the following function:

$$\Phi_k = f(\mathbf{x}_k) + \eta\Delta_k^2, \tag{2.14}$$

where η satisfies:

$$0 < \eta < \gamma(1-\delta)^2. \tag{2.15}$$

Note that, whenever Assumption 2.1 holds, we have

$$\Phi_k \geq f_{\min}. \tag{2.16}$$

Function Φ_k allows us to state the following result which characterizes the evolution of the algorithm at each iteration.

Proposition 2.3.3. *Let $\{\mathbf{x}_k\}$ and $\{\tilde{\alpha}_k^i\}$, $i = 1,\dots,n$, be the sequences produced by* `LAM` *using* `Unconstrained DF-Linesearch`*. Then for all $k = 0, 1, \dots$:*

$$\Phi_{k+1} - \Phi_k \leq -\tilde{c}_{LAM}\Delta_{k+1}^2, \tag{2.17}$$

where

$$\tilde{c}_{LAM} \;=\; \min\left\{\eta\left(\frac{1-\theta^2}{\theta^2}\right), \gamma c^2, \left(\gamma(1-\delta)^2 - \eta\right)\right\}, \tag{2.18}$$

η is a parameter satisfying (2.15) *and $c, \gamma, \delta, \theta$ are the constants defined in the* `LAM` *algorithm and the* `Unconstrained DF-Linesearch` *procedure.*

Proof. For each $k \geq 1$, consider the iteration $k-1$. We split the set of iteration indices $\{0, 1, 2, \dots\}$ into the two subsets $\mathcal{K}_{\mathtt{s}}$ and $\mathcal{K}_{\mathtt{u}}$, namely

i) $k \in \mathcal{K}_{\mathtt{s}}$ when $x_{k+1} \neq x_k$;

ii) $k \in \mathcal{K}_{\mathtt{u}}$ when $x_{k+1} = x_k$.

Then the following cases can occur.

- $\mathbf{x}_k \neq \mathbf{x}_{k-1}$ (i.e., $k-1 \in \mathcal{K}_{\mathtt{s}}$) and $\Delta_k = \Delta_{k-1}$. Hence, there exists a coordinate $\bar{\imath}$ such that we have moved from $\mathbf{x}_{k-1}$ along $\pm e_{\bar{\imath}}$ and we have

  $$\alpha_{k-1}^{\bar{\imath}} \geq \nu_{k-1}^{\bar{\imath}} = \max\left\{\tilde{\alpha}_{k-1}^{\bar{\imath}}, c\Delta_{k-1}\right\} \geq c\Delta_{k-1} = c\Delta_k.$$

  Then, from the line-search procedure, we can write

  $$f(\mathbf{x}_k) \leq f(\mathbf{x}_{k-1}) - \gamma c^2\Delta_k^2.$$

  Hence, we have

  $$\Phi_k - \Phi_{k-1} = f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \leq -\gamma c^2\Delta_k^2. \tag{2.19}$$

- $\mathbf{x}_k \neq \mathbf{x}_{k-1}$ (i.e., $k-1 \in \mathcal{K}_{\mathbf{s}}$) and $\Delta_k > \Delta_{k-1}$. Then we have that an index $\bar{\jmath}$ exists such that a line-search has been performed along the $\bar{\jmath}$-th direction producing the step length $\alpha_{k-1}^{\bar{\jmath}}$ satisfying
$$\alpha_{k-1}^{\bar{\jmath}} = \tilde{\alpha}_k^{\bar{\jmath}} = \Delta_k.$$
More specifically, we have
$$\begin{aligned} f(\mathbf{x}_k) &\leq f(\mathbf{y}_{k-1}^{\bar{\jmath}} + \tilde{\alpha}_k^{\bar{\jmath}} d_k^{\bar{\jmath}}) \\ &\leq f(\mathbf{y}_{k-1}^{\bar{\jmath}} + \delta\tilde{\alpha}_k^{\bar{\jmath}} d_k^{\bar{\jmath}}) - \gamma(1-\delta)^2(\tilde{\alpha}_k^{\bar{\jmath}})^2 \leq f(\mathbf{x}_{k-1}) - \gamma(1-\delta)^2(\tilde{\alpha}_k^{\bar{\jmath}})^2 \\ &= f(x_{k-1}) - \gamma(1-\delta)^2\Delta_k^2. \end{aligned} \tag{2.20}$$
Hence, we obtain
$$\begin{aligned} \Phi_k - \Phi_{k-1} \quad &= f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \\ &\leq -\gamma(1-\delta)^2\Delta_k^2 + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \\ &\leq -\gamma(1-\delta)^2\Delta_k^2 + \eta\Delta_k^2 \\ &\leq -\left(\gamma(1-\delta)^2 - \eta\right)\Delta_k^2. \end{aligned} \tag{2.21}$$

- $x_k = x_{k-1}$ (i.e., $k-1 \in \mathcal{K}_{\mathbf{u}}$). Recalling Lemma 2.3.1, we have
$$\Delta_k = \theta\Delta_{k-1}.$$
Then,
$$\begin{aligned} \Phi_k - \Phi_{k-1} &= f(x_k) - f(x_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \\ &\leq \eta\left(\Delta_k^2 - \frac{1}{\theta^2}\Delta_k^2\right) \\ &= -\eta\left(\frac{1-\theta^2}{\theta^2}\right)\Delta_k^2. \end{aligned} \tag{2.22}$$

Finally (2.22), (2.19) and (2.21) conclude the proof. ☐

Exploiting the properties of Φ_k, we prove that the sequences of initial step-sizes $\tilde{\alpha}_k^i$, $i = 1, \dots, n$, are all convergent to zero.

Proposition 2.3.4. *Suppose that Assumption 2.1 holds. Then,* `LAM` *using* `Unconstrained DF-Linesearch` *produces sequences* $\{\tilde{\alpha}_k^i\}$, $i = 1, \dots, n$, *such that*
$$\lim_{k\to\infty} \max_{i=1,\dots,n} \{\tilde{\alpha}_k^i\} = 0.$$

Proof. By Proposition 2.3.3, the sequence $\{\Phi_k\}$ is monotonically decreasing. Since $\Phi_k \geq f_{\min}$, we have that
$$\lim_{k\to\infty} \Phi_k = \bar{\Phi}.$$
Recalling again Proposition 2.3.3 and the above limit, we get
$$\lim_{k\to\infty} \Delta_k = 0,$$
and using the definition of Δ_k the proof is concluded. ☐

Corollary 2.3.5. *Suppose that Assumption 2.1 holds. Then,* `LAM` *using* `Unconstrained DF-Linesearch` *produces an infinite sequence* $\{x_k\}$ *such that*

$$\lim_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| = 0.$$

Proof. The proof easily follows recalling Proposition 2.3.2 and Proposition 2.3.4. □

Remark *Note that the result of Proposition 2.3.4 is somewhat stronger than analogous results for GPS [85] and MADS-type [4] algorithms. Indeed, for those algorithms it is only possible to show that a subsequence of the step-sizes converges to zero. As a consequence, also the result of Corollary 2.3.5 is stronger in that it states that every limit point of the sequence of iterates is stationary.*

2.3.3 Complexity bounds for LAM

This section is devoted to the definition of the worst case complexity bounds for the `LAM` algorithm. The next two propositions ensure that the algorithm model takes at most $\mathcal{O}(n\epsilon^{-2})$ iterations and $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations to produce a point $\mathbf{x}_k$ such that $\|\nabla f(\mathbf{x}_k)\| \le \epsilon$.

Proposition 2.3.6. *Suppose that Assumption 2.1 holds. Let* $\{\mathbf{x}_k\}$ *be the sequence of points produced by* `LAM` *using* `Unconstrained DF-Linesearch`. *Given any* $\epsilon \in (0,1)$ *and* $\tilde\alpha_0^i \ge \epsilon$, *for* $i = 1,\dots,n$, *assume that* $\bar\jmath_\epsilon + 1$ *is the first iteration such that* $\|\nabla f(\mathbf{x}_{\bar\jmath_\epsilon+1})\| \le \epsilon$, *i.e.* $\|\nabla f(\mathbf{x}_k)\| > \epsilon$, *for all* $k = 0,1,\dots,\bar\jmath_\epsilon$. *Then,*

$$\bar\jmath_\epsilon \le \left\lfloor \frac{n\ c_1^2\ (\Phi_0 - f_{\min})}{\tilde c_{LAM}}\ \epsilon^{-2} \right\rfloor = \mathcal{O}(n\epsilon^{-2}), \tag{2.23}$$

where $\tilde c_{LAM}$ *is given by (2.18) and*

$$c_1 \quad = \quad \frac{\gamma + L}{\theta}. \tag{2.24}$$

Proof. Using the function Φ_k defined by (2.14) we can write:

$$\Phi_{\bar\jmath+1} - \Phi_0 = (\Phi_{\bar\jmath+1} - \Phi_{\bar\jmath}) + (\Phi_{\bar\jmath} - \Phi_{\bar\jmath-1}) + \cdots + (\Phi_1 - \Phi_0)$$

and exploiting Proposition 2.3.3 we have:

$$\Phi_{\bar\jmath+1} - \Phi_0 \le -\tilde c_{\text{LAM}} \sum_{k=1}^{\bar\jmath+1} \Delta_k^2 = -\tilde c_{\text{LAM}} \sum_{k=0}^{\bar\jmath} \Delta_{k+1}^2.$$

By recalling (2.16), we can write

$$f_{\min} - \Phi_0 \le \Phi_{\bar\jmath} - \Phi_0 \le -\tilde c_{\text{LAM}} \sum_{k=0}^{\bar\jmath} \Delta_{k+1}^2. \tag{2.25}$$

As done previously the set of iteration indices $\{0,1,2,\dots\}$ can be divided into the two subsets $\mathcal{K}_{\mathrm{s}}$ and $\mathcal{K}_{\mathrm{u}}$, namely

i) $k \in \mathcal{K}_{\mathrm{s}}$ when $x_{k+1} \ne x_k$;

ii) $k \in \mathcal{K}_{\mathrm{u}}$ when $x_{k+1} = x_k$.

Furthermore Proposition 2.3.4 implies that $\mathcal{K}_{\mathtt{u}}$ is infinite.
Now it is possible to define the following two set of indices:

$$\begin{aligned} \mathcal{J}_{\mathtt{s}} &= \{0, \dots, \bar{\jmath}\} \cap \mathcal{K}_{\mathtt{s}} \quad \text{succ. iterations up to } \bar{\jmath}, \\ \mathcal{J}_{\mathtt{u}} &= \{0, \dots, \bar{\jmath}\} \cap \mathcal{K}_{\mathtt{u}} \quad \text{unsucc. iterations up to } \bar{\jmath}, \end{aligned}$$

and to rewrite (2.25):

$$f_{\min} - \Phi_0 \le -\tilde{c}_{\text{LAM}} \sum_{k \in \mathcal{J}_{\mathtt{s}}} \Delta_{k+1}^2 - \tilde{c}_{\text{LAM}} \sum_{k \in \mathcal{J}_{\mathtt{u}}} \Delta_{k+1}^2. \tag{2.26}$$

For all $k \in \mathcal{J}_{\mathtt{s}}$ we associate an index $m(k)$ given by:

i) if $\mathcal{J}_{\mathtt{u}} \cap \{0, \dots, k-1\} \neq \emptyset$ then $m(k)$ in the biggest index of $\mathcal{J}_{\mathtt{u}} \cap \{0, \dots, k-1\}$;

ii) if $\mathcal{J}_{\mathtt{u}} \cap \{0, \dots, k-1\} = \emptyset$ then $m(k) = -1$.

The steps of `LAM` ensure that, for all $k \in \mathcal{J}_{\mathtt{s}}$,

$$\Delta_{k+1}^2 \ge \Delta_{m(k)+1}^2. \tag{2.27}$$

Using these inequalities in (2.26), we obtain

$$f_{\min} - \Phi_0 \le -\tilde{c}_{\text{LAM}} \sum_{k \in \mathcal{J}_s} \Delta_{m(k)+1}^2 - \tilde{c}_{\text{LAM}} \sum_{k \in \mathcal{J}_u} \Delta_{k+1}^2. \tag{2.28}$$

Now recalling (2.4) of Proposition 2.3.2, the choices for α_0^i, for $i = 1, \dots, n$ and that $\|\nabla f(\mathbf{x}_k)\| > \epsilon$, for all $k = 0, 1, \dots, \bar{\jmath}_\epsilon$, we obtain:

$$\Phi_0 - f_{\min} \ge (\bar{\jmath} + 1)\tilde{c}_{\text{LAM}} \frac{\theta^2}{n(\gamma + L)^2} \epsilon^2.$$

Thus, the number $\bar{\jmath}_\epsilon$ of iterations can be bounded from above by

$$\bar{\jmath}_\epsilon \le \left\lfloor \frac{n(\gamma + L)^2 (\Phi_0 - f_{\min})}{\tilde{c}_{\text{LAM}}\ \theta^2} \epsilon^{-2} \right\rfloor = \mathcal{O}(n\epsilon^{-2})$$

which concludes the proof. ☐

Now, we prove the worst case complexity bound for the number of function evaluations.

Proposition 2.3.7. *Suppose that Assumption 2.1 holds. Let $\{\mathbf{x}_k\}$ be the sequence of point produced by LAM using* `Unconstrained DF-Linesearch`*. Given any $\epsilon \in (0, 1)$ and $\tilde{\alpha}_0^i \ge \epsilon$, assume that $\bar{\jmath}_\epsilon + 1$ is the first iteration such that $\|\nabla f(\mathbf{x}_{\bar{\jmath}_\epsilon + 1})\| \le \epsilon$, i.e. $\|\nabla f(\mathbf{x}_k)\| > \epsilon$, for all $k = 0, 1, \dots, \bar{\jmath}_\epsilon$. Then, the number of function evaluations $\#f_\epsilon$ required by* `LAM` *until the $\bar{\jmath}_\epsilon$-th iteration are in the worst case such that*

$$\begin{aligned} \#f_\epsilon \le 2n &\left\lfloor \frac{n c_1^2 (\Phi_0 - f_{\min})}{\tilde{c}_{LAM}} \epsilon^{-2} \right\rfloor + \\ &\left\lfloor \frac{n c_1^2 (f(x_0) - f_{\min})}{\gamma c^2} \max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\} \epsilon^{-2} \right\rfloor = \mathcal{O}(n^2 \epsilon^{-2}), \end{aligned}$$

where c_1 and $\tilde{c}_{LAM}$ are defined in (2.24) and (2.18), respectively.

Proof. By assumption, for all $k = 0, 1, \dots, \bar{\jmath}$, we have that

$$\epsilon < \|\nabla f(\mathbf{x}_k)\|. \tag{2.29}$$

Let $\mathcal{U}_{\bar{\jmath}_\epsilon}$ and $\mathcal{S}_{\bar{\jmath}_\epsilon}$ be the index sets of unsuccessful and successful iterations until the iteration $\bar{\jmath}_\epsilon$.

Then, for every iteration k, if $k \in \mathcal{U}_{\bar{\jmath}_\epsilon}$, the Algorithm performs

$$\#f_k^u = 2n,$$

function evaluations.

On the other hand, if $k \in \mathcal{S}_{\bar{\jmath}_\epsilon}$, we can partition the function evaluations performed by the algorithm into those producing a sufficient decrease in the objective function and those producing a failure, i.e. the last function evaluation performed by the `Unconstrained DF-Linesearch` procedure. Hence, we divide the number of such function evaluations into $\#f^s$ and $\overline{\#f}^s$. For the latter, we have

$$\overline{\#f}^s \leq 2n|\mathcal{S}_{\bar{\jmath}_\epsilon}|.$$

Exploiting the previous inequality, we can write

$$\#f_\epsilon = \#f_k^u|\mathcal{U}_{\bar{\jmath}_\epsilon}| + \overline{\#f}^s + \#f^s \leq 2n\left(|\mathcal{U}_{\bar{\jmath}_\epsilon}| + |\mathcal{S}_{\bar{\jmath}_\epsilon}|\right) + \#f^s = 2n\bar{\jmath}_\epsilon + \#f^s \tag{2.30}$$

Concerning $\#f^s$, each time that such a function evaluation is performed, we have, by the instructions of `Unconstrained DF-Linesearch`, either

$$f(\mathbf{y}_k^i) - f(\mathbf{y}_k^i + \alpha_j\hat{\mathbf{d}}) \geq \gamma\alpha_j^2 \geq \gamma \min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\} c^2\Delta_k^2,$$

or

$$f(\mathbf{y}_k^i + \alpha_j\hat{\mathbf{d}}) - f(\mathbf{y}_k^i + \alpha_j/\delta\hat{\mathbf{d}}) \geq \gamma\left(\frac{1-\delta}{\delta}\right)^2\alpha_j^2 \geq \gamma + \alpha_j\hat{\mathbf{d}}) \geq \gamma\alpha_j^2 \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\} c^2\Delta_k^2.$$

Now, for all $k \in \mathcal{S}_{\bar{\jmath}_\epsilon}$ we can define an index $\tilde{m}(k)$ given by:

i) if $k > 0$ and $\mathcal{U}_{\bar{\jmath}_\epsilon} \cap \{0, \dots, k-1\} \neq \emptyset$ then $\tilde{m}(k)$ in the biggest index of $\mathcal{U}_{\bar{\jmath}_\epsilon} \cap \{0, \dots, k-1\}$;

ii) if $k = 0$ or $\mathcal{U}_{\bar{\jmath}_\epsilon} \cap \{0, \dots, k-1\} = \emptyset$ then $\tilde{m}(k) = 0$.

Then, we obtain:

$$f(\mathbf{y}_k^i + \alpha_j\hat{\mathbf{d}}) - f(\mathbf{y}_k^i + \alpha_j/\delta\hat{\mathbf{d}}) \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\} c^2\Delta_{\tilde{m}(k)}^2 \geq \gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\} c^2\frac{\epsilon^2}{nc_1^2}.$$

Then, recalling that f is bounded from below by $f_{\min}$, summing the above relation over all such function evaluations, we obtain

$$f_0 - f_{\min} \geq \#f^s\gamma\min\left\{1, \left(\frac{1-\delta}{\delta}\right)^2\right\} c^2\frac{\epsilon^2}{nc_1^2},$$

so that

$$\#f^s \leq \frac{c_1^2 n(f_0 - f_{\min})}{\gamma c^2 \epsilon^2} \max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\}.$$

Finally, using (2.30) and (2.23), the following holds

$$\#f_\epsilon \leq 2n \left\lfloor \frac{nc_1^2(\Phi_0 - f_{\min})}{\tilde{c}_{\text{LAM}}} \epsilon^{-2} \right\rfloor + \left\lfloor \frac{nc_1^2(f_0 - f_{\min})}{\gamma c^2 \epsilon^2} \max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\} \right\rfloor = \mathcal{O}(n^2 \epsilon^{-2}), \tag{2.31}$$

where c_1 and $\tilde{c}_{\text{LAM}}$ are defined in (2.24) and (2.18), respectively. The latter concludes the proof. ☐

The previous results show that the line-search DFO approach is also able to propose algorithms with exactly the same complexity bounds of those obtained in [86] for direct search methods.
In addition, the next proposition shows that the use of the line-search technique allows us to guarantee to a DFO algorithm the further property that the number of iterations such that $\|\nabla f(\mathbf{x}_k)\| > \epsilon$ is of the order of ϵ^{-2}. In particular, for the `LAM` algorithm, the number of such iterations is $\mathcal{O}(n^2\ \epsilon^{-2})$ in the worst case.

Proposition 2.3.8. *Suppose that Assumption 2.1 holds. Let $\{x_k\}$ be the sequence of point produced by `LAM` using `Unconstrained DF-Linesearch`. Given any $\epsilon \in (0,1)$, consider the following subset of indices:*

$$\mathcal{K}_\epsilon = \{\ k = 1, \dots :\ \|\nabla f(\mathbf{x}_k)\| > \epsilon\ \}. \tag{2.32}$$

Then,

$$|\mathcal{K}_\epsilon| \leq \left\lfloor \frac{n\ c_2^2\ (\Phi_0 - f_{\min})}{\tilde{c}_{LAM}}\ \epsilon^{-2} \right\rfloor = \mathcal{O}(n^2\epsilon^{-2}), \tag{2.33}$$

where $\tilde{c}_{LAM}$ is given by (2.18) and

$$c_2 \quad = \quad \max\left\{ \left(\frac{\gamma + L(\sqrt{n}+1)}{\delta}\right), \left(\frac{\gamma + L}{\theta}\right) \right\}. \tag{2.34}$$

Proof. Proposition 2.3.3 shows that the sequence $\{\Phi_k\}$ is not increasing and that, for every k, we have:

$$\Phi_k - \Phi_0 \leq -\tilde{c}_{\text{LAM}} \sum_{\tilde{k}=1}^{k} \Delta_{\tilde{k}}^2 = -\tilde{c}_{\text{LAM}} \sum_{\tilde{k}=0}^{k} \Delta_{\tilde{k}+1}^2. \tag{2.35}$$

Since the sequence $\{\Phi_k\}$ is bounded from below, Φ^* exists such that:

$$\lim_{k\to\infty} \Phi_k = \Phi^* \geq f_{\min}.$$

Taking the limit for $k \to \infty$ in (2.35) we obtain:

$$\Phi_0 - \Phi^* \geq \tilde{c}_{\text{LAM}} \sum_{k=0}^{\infty} \Delta_{k+1}^2 \geq \tilde{c}_{\text{LAM}} \sum_{k\in\mathcal{K}_\epsilon} \Delta_{k+1}^2,$$

which, together with the definition of $\mathcal{K}_\epsilon$ and Proposition 2.3.2, gives

$$\Phi_0 - f_{\min} \geq \tilde{c}_{\text{LAM}} \sum_{k\in\mathcal{K}_\epsilon} \Delta_{k+1}^2 \geq |\mathcal{K}_\epsilon| \tilde{c}_{\text{LAM}} \frac{\epsilon^2}{nc_2^2}.$$

Thus

$$|\mathcal{K}_\epsilon| \leq \left\lfloor \frac{nc_2^2(\Phi_0 - f_{\min})}{\tilde{c}_{\text{LAM}}} \epsilon^{-2} \right\rfloor = \mathcal{O}(n^2\epsilon^{-2})$$

and the proof is concluded. ☐

2.4 Bound-constrained problems

In this section we see the insights of the framework proposed for bound-constrained problems. We start again by introducing the line-search within the algorithm in such case and describe its main features. We then see some preliminary results that are used afterwards to prove the convergence properties of `LAM`, and we prove the worst-case complexity bounds in terms of iterations and function evaluations. Finally we analyze the properties of the method regarding the identification of the active set.

2.4.1 DF-Linesearch with bound constraints

In the following we describe the line-search procedure for bound-constrained problems.

Algorithm 8 Bounded DF-Linesearch($\mathbf{x}$,i,γ,δ,ν)

```
 1: if $\nu > \max\{u_i - x_i, x_i - \ell_i\}$ then return $\mathbf{d} = \mathbf{e}_i$, $\alpha = 0$
 2: end if
 3: set $\bar{\alpha} = \nu$
 4: if $\bar{\alpha} \leq x_i - l_i$ and $f(\mathbf{x} - \bar{\alpha}\mathbf{e}_i) \leq f(\mathbf{x}) - \gamma\bar{\alpha}^2$ then
 5:     set $\mathbf{d} = -\mathbf{e}_i$, $\alpha_{\max} = x_i - l_i$ and go to line 12
 6: end if
 7: if $\bar{\alpha} \leq u_i - x_i$ and $f(\mathbf{x} + \bar{\alpha}\mathbf{e}_i) \leq f(\mathbf{x}) - \gamma\bar{\alpha}^2$ then
 8:     set $\mathbf{d} = \mathbf{e}_i$, $\alpha_{\max} = u_i - x_i$ and go to line 12
 9: else
10:     return $\mathbf{d}$ and $\alpha = 0$
11: end if
12: set $\alpha = \bar{\alpha}$ and $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$
13: while $(\alpha < \alpha_{\max}$ and $f(\mathbf{x} + \omega\mathbf{d}) \leq f(\mathbf{x} + \alpha\mathbf{d}) - \gamma(\omega - \alpha)^2)$ do
14:     set $\alpha = \omega$ and $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$
15: end while
16: return $\mathbf{d}$ and $\alpha$
```

Given a feasible point $\mathbf{x}$, the exploration of the i-th coordinate direction $\mathbf{e}_i$ is performed by a line-search procedure outlined in the `Bounded DF-Linesearch`. First, we check if the given step-size ν is feasible along $\pm\mathbf{e}_i$, that is, if one among $\mathbf{x} + \nu\mathbf{e}_i$ and $\mathbf{x} - \nu\mathbf{e}_i$ is feasible. If that is not the case, we end the line-search returning a zero step length to indicate a failure. Otherwise, we try to determine if one between $\mathbf{e}_i$ and $-\mathbf{e}_i$ is a “good” descent direction, that is, if a sufficient decrease in the objective function can be obtained by using a feasible step-size, with the same rule used for the unconstrained case. If neither $\mathbf{e}_i$ nor $-\mathbf{e}_i$ qualifies as a suitable descent direction, then the line-search procedure terminates, returning a zero step length to indicate a failure. If a sufficient decrease of f is achieved, then an *extrapolation* (or *expansion*) phase starts (i.e., lines 12–15), where we try to increase the step-size to the maximum extent while preserving feasibility and guaranteeing the sufficient decrease condition. Specifically, the *while* loop keeps expanding the step-size as long as the most recently accepted point remains strictly within the bounds (i.e., $\alpha < \alpha_{\max}$) and the new tentative point is sufficiently better than the last accepted one (i.e., $f(\mathbf{x} + \omega\mathbf{d}) \leq f(\mathbf{x} + \alpha\mathbf{d}) - \gamma(\omega - \alpha)^2$).

For the seek of clarity, let us briefly compare the `Bounded DF-Linesearch` with the `Unconstrained DF-Linesearch`. The structure both the algorithms is the same. First, the methods verify wether the sufficient decrease condition is satisfied along one of the sides of the current coordinate direction, performing a step of the given size ν. If the condition is met, then the expansion procedure begins, otherwise the algorithms stop returning a step-size equal to zero. The difference lies in the fact that, whenever the `Bounded DF-Linesearch` tests the sufficient condition, it also checks if the trial point is feasible. It is important to point out that, in the first phase, the `Bounded DF-Linesearch` is immediately terminated if, along both sides of the current direction, the trial points are found infeasible. On the contrary, during the extrapolation phase, if a trial point is found infeasible, it is projected on the bound, and the sufficient decrease condition is verified for the the latter.

As in the previous section, we state the following property of `LAM`. Note that the same arguments used for Lemma 2.3.1 in the unconstrained case can be used in the following Lemma, so the proof is omitted.

Lemma 2.4.1. *Let $\{\Delta_k\}$ be the sequence of maximum tentative step-sizes produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Then, for all k we have*

$$\Delta_{k+1} \begin{cases} \geq \Delta_k & \text{if } \mathbf{x}_{k+1} \neq \mathbf{x}_k \\ = \theta \Delta_k & \text{if } \mathbf{x}_{k+1} = \mathbf{x}_k. \end{cases}$$

Therefore, $\Delta_k \leq \Delta_{k+1}/\theta$ for all k.

2.4.2 Preliminary results

In this section we consider (P1) in the case where $\Omega = [\mathbf{l}, \mathbf{u}]$.

The following assumption is considered satisfied throughout this section, even when not explicitly invoked.

Assumption 2.2. *A constant $M_g \geq 0$ exists such that*

$$\|\nabla f(\mathbf{x})\| \leq M_g$$

for all $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$.

Under Assumption 2.1, we can also provide the following definition of coordinate-wise Lipschitz constants.

Definition 2.4.2. *The coordinate-wise Lipschitz constants $L_i > 0$, $i = 1, \dots, n$, of ∇f are such that, for all $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$,*

$$|\nabla_i f(\mathbf{x} + s\mathbf{e}_i) - \nabla_i f(\mathbf{x})| \leq L_i |s| \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}], \quad i = 1, \dots, n.$$

Moreover,

$$L^{max} = \max_{i=1,\dots,n} L_i. \tag{2.36}$$

Now, given $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$, let us introduce the following criticality measure:

$$\chi(\mathbf{x}) = \max_{\substack{\mathbf{x}+\mathbf{d} \in [\mathbf{l}, \mathbf{u}] \\ \|\mathbf{d}\| \leq 1}} -\nabla f(\mathbf{x})^\top \mathbf{d}.$$

The above measure has been successfully used in the analysis of some direct search methods for linearly constrained problems [51, 60, 61]. It can be interpreted as the progress on a first-order model in a ball centered at x with unit ray subject to feasibility constraints [29, 61], thus generalizing $\|\nabla f(\mathbf{x})\|$ from the unconstrained setting. Originally proposed in [28] for more general constraints and further analyzed in [29], $\chi(\mathbf{x})$ is continuous, non-negative and such that $\chi(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ is a KKT point. So, we can define a stationary point as follows.

Definition 2.4.3. *A point* $\mathbf{x}^* \in [\mathbf{l}, \mathbf{u}]$ *is said to be a stationary point of problem* (P1) *if* $\chi(\mathbf{x}^*) = 0$.

Next, given $\epsilon \geq 0$, we define the set of ϵ-active constraints at $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$ as

$$\mathcal{I}_l(\mathbf{x}, \epsilon) = \{i : x_i \leq l_i + \epsilon\},$$
$$\mathcal{I}_u(\mathbf{x}, \epsilon) = \{i : x_i \geq u_i - \epsilon\}.$$

Namely, $\mathcal{I}_l(\mathbf{x}, \epsilon)$ and $\mathcal{I}_u(\mathbf{x}, \epsilon)$ denote the sets of lower and upper bound constraints, respectively, that are nearly active at $\mathbf{x}$ with a tolerance ϵ. Accordingly, let us define $N(\mathbf{x}, \epsilon)$ as the ϵ-normal cone generated by the ϵ-active constraints, that is,

$$N(\mathbf{x}, \epsilon) = \operatorname{cone}\Big(\{-\mathbf{e}_i, i \in \mathcal{I}_l(\mathbf{x}, \epsilon)\} \cup \{\mathbf{e}_i, i \in \mathcal{I}_u(\mathbf{x}, \epsilon)\} \cup \{0\}\Big),$$

while the ϵ-tangent cone $T(\mathbf{x}, \epsilon)$ is the polar of $N(\mathbf{x}, \epsilon)$, that is,

$$T(\mathbf{x}, \epsilon) = N(\mathbf{x}, \epsilon)^\circ = \{\mathbf{d} \in \mathbb{R}^n : \mathbf{d}^\top \mathbf{v} \leq 0, \ \forall \ \mathbf{v} \in N(\mathbf{x}, \epsilon)\}.$$

The use of ϵ-normal and ϵ-tangent cones is a well known tool in the analysis of direct search methods applied to linearly constrained problems [51, 60, 61]. Essentially, the set $\mathbf{x} + T(\mathbf{x}, \epsilon)$ is an approximation of the feasible region near a feasible point $\mathbf{x}$, that is, moving from $\mathbf{x}$ along any direction in $T(\mathbf{x}, \epsilon)$ with a step-size less than or equal to ϵ ensures that all constraints stay satisfied.

In our case, considering the structure of the feasible set of problem (P1), it is straightforward to verify that a set of generators for $T(\mathbf{x}, \epsilon)$ is given by

$$G_{T(\mathbf{x},\epsilon)} = \{-\mathbf{e}_i, i \notin \mathcal{I}_l(\mathbf{x}, \epsilon)\} \cup \{\mathbf{e}_i, i \notin \mathcal{I}_u(\mathbf{x}, \epsilon)\} \cup \{0\}, \tag{2.37}$$

that is, $T(\mathbf{x}, \epsilon) = \operatorname{cone}(G_{T(\mathbf{x},\epsilon)})$.

The following two propositions from [61] show how $\chi(\mathbf{x}_k)$ can be bounded above by using the projection of $-\nabla f(\mathbf{x}_k)$ onto $T(\mathbf{x}_k, \epsilon)$ and $N(\mathbf{x}_k, \epsilon)$.

Proposition 2.4.4 ([60, Proposition 8.2])**.** *If* $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$*, then for all* $\epsilon \geq 0$ *we have that*

$$\chi(\mathbf{x}) \leq \|(-\nabla f(\mathbf{x}))_{T(\mathbf{x},\epsilon)}\| + \epsilon\sqrt{n}\|(-\nabla f(\mathbf{x}))_{N(\mathbf{x},\epsilon)}\|.$$

Proposition 2.4.5 ([60, Proposition 8.1])**.** *Given* $\epsilon \geq 0$*, let* $G_{T(\mathbf{x},\epsilon)}$ *be defined as in* (2.37)*. If* $(-\nabla f(\mathbf{x}))_{T(\mathbf{x},\epsilon)} \neq 0$*, then there exists* $\mathbf{d} \in G_{T(\mathbf{x},\epsilon)}$ *such that*

$$\frac{1}{\sqrt{n}}\|(-\nabla f(\mathbf{x}))_{T(\mathbf{x},\epsilon)}\| \leq -\nabla f(\mathbf{x})^\top \mathbf{d}.$$

2.4.3 Global convergence

In the following proposition, we show that `LAM` using the `Bounded DF-Linesearch` is well defined, i.e., the `Bounded DF-Linesearch` cannot cycle so that the sequences of points and step-sizes produced are infinite overall.

Proposition 2.4.6. *`LAM` using the `Bounded DF-Linesearch` is well defined, i.e., it produces infinite sequences* $\{\mathbf{x}_k\}$, $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, $i = 1, \dots, n$.

Proof. We have to show that the `Bounded DF-Linesearch` procedure cannot infinitely cycle over steps 12–14 in the while loop. Let us suppose, by contradiction, that the while loop does not terminate, i.e., we always have $\alpha < \alpha_{\max}$. At the j-th iteration of the while loop, we have

$$\alpha = \frac{\bar{\alpha}}{\delta^j}$$

with $\delta < 1$. If $\alpha_{\max}$ is finite, then we have $\alpha > \alpha_{\max}$, for j sufficiently large, which is a contradiction. Otherwise, if $\alpha_{\max} = +\infty$, then, for every j we have

$$f(\mathbf{x} + \omega d) \le f(\mathbf{x} + \alpha \mathbf{d}) - \gamma(\omega - \alpha)^2, \tag{2.38}$$

with $\omega = \alpha/\delta = \bar{\alpha}/\delta^{j+1}$. Now, for j sufficiently large, (2.38) contradicts Assumption 2.1, i.e., that f is bounded from below inside the feasible set. ☐

In the following, we present some results concerning the global convergence of `LAM` using the `Bounded DF-Linesearch` to stationary points. More specifically, by extending some results from [74] and Section 2.3, we establish a relationship between $\nabla f(\mathbf{x}_k)$, with specific directions $\mathbf{d}$, and the largest tentative step length at Δ_{k+1}. As to be shown, the bound depends, besides on the problem dimension n, on the Lipschitz constant L and the algorithm parameters γ, θ and δ.

From now on, let us denote

$$T_k := T(\mathbf{x}_k, \Delta_k) \quad \text{and} \quad G_k := G_{T(\mathbf{x}_k, \Delta_k)}, \tag{2.39}$$

where G_k is the set of generators of T_k and is defined as in (2.37).

Theorem 2.4.7. *Let $\{\mathbf{x}_k\}$ be the sequence produced by `LAM` using the `Bounded DF-Linesearch`. Then, for all k and for all $\mathbf{d} \in G_k$, we have that*

$$-\nabla f(\mathbf{x}_k)^\top \mathbf{d} \le \begin{cases} \left(\dfrac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1} & \text{if } \mathbf{x}_{k+1} \neq \mathbf{x}_k, \\ \left(\dfrac{\gamma + L^{max}}{\theta}\right)\Delta_{k+1} & \text{if } \mathbf{x}_{k+1} = \mathbf{x}_k. \end{cases} \tag{2.40}$$

Proof. First, consider an iteration k such that $\mathbf{x}_{k+1} \neq \mathbf{x}_k$ (i.e., a successful iteration). The following cases can occur, recalling that the analysis is limited to considering directions $\pm\mathbf{e}_i$, $i = 1, \dots, n$, belonging to G_k.

(1a) $\mathbf{e}_i \in G_k$, $(\mathbf{y}_k^{i+1})_i = l_i$ and $(\mathbf{y}_k^i)_i = l_i$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. From the instructions of the line-search procedure, we have that

$$f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) > f(\mathbf{y}_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2.$$

By the mean value theorem, we have

$$f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - f(\mathbf{y}_k^i) = \tilde{\alpha}_{k+1}^i \nabla_i f(\boldsymbol{\xi}_k^i),$$

where $\boldsymbol{\xi}_k^i = \mathbf{y}_k^i + t_k^i \tilde{\alpha}_{k+1}^i \mathbf{e}_i$ and $t_k^i \in (0, 1)$. Then,

$$-\nabla_i f(\boldsymbol{\xi}_k^i) < \gamma \tilde{\alpha}_{k+1}^i.$$

It follows that

$$-\nabla_i f(\boldsymbol{\xi}_k^i) + \nabla_i f(\mathbf{x}_k) - \nabla_i f(\mathbf{x}_k) < \gamma \tilde{\alpha}_{k+1}^i$$

and we can write

$$\begin{aligned} -\nabla_i f(\mathbf{x}_k) &< \gamma \tilde{\alpha}_{k+1}^i - (\nabla_i f(\mathbf{x}_k) - \nabla_i f(\boldsymbol{\xi}_k^i)) \\ &\le \gamma \tilde{\alpha}_{k+1}^i + L\|\mathbf{x}_k - \boldsymbol{\xi}_k^i\| \\ &\le \gamma \tilde{\alpha}_{k+1}^i + L\|\mathbf{x}_k - \mathbf{y}_k^i\| + L\|\mathbf{y}_k^i - \boldsymbol{\xi}_k^i\|. \end{aligned} \tag{2.41}$$

Moreover, since $t_k^i \in (0,1)$, we have $t_k^i \tilde{\alpha}_{k+1}^i \le \tilde{\alpha}_{k+1}^i$. Then,

$$\|\mathbf{y}_k^i - \boldsymbol{\xi}_k^i\| \le \tilde{\alpha}_{k+1}^i.$$

Hence, since $\alpha_{k+1}^i \le \tilde{\alpha}_{k+1}^i \le \Delta_{k+1}$ for all $i = 1, \dots, n$, and taking into account that $\mathbf{y}_k^i = \mathbf{x}_k + \sum_{j=1}^{i-1} \alpha_k^j \mathbf{d}_k^j$, so that $\|\mathbf{x}_k - \mathbf{y}_k^i\| \le \sqrt{n}\Delta_{k+1}$, we get

$$(-\nabla f(\mathbf{x}_k))^\top \mathbf{e}_i < \gamma \tilde{\alpha}_{k+1}^i + L\|\mathbf{x}_k - \mathbf{y}_k^i\| + L\tilde{\alpha}_{k+1}^i \le \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1}. \tag{2.42}$$

(1b) $-\mathbf{e}_i \in G_k$, $(\mathbf{y}_k^{i+1})_i = l_i$ and $(\mathbf{y}_k^i)_i > l_i$. Then, $\tilde{\alpha}_{k+1}^i = \alpha_k^i > 0$. From the instructions of the line-search procedure, there exist β_k^i and $\overline{\beta_k^i}$ such that $\beta_k^i \in [\nu_k^i, \tilde{\alpha}_{k+1}^i]$, $\overline{\beta_k^i} \in \{0\} \cup [\nu_k^i, \tilde{\alpha}_{k+1}^i)$ and $\beta_k^i - \overline{\beta_k^i} > 0$, and the following holds

$$f(\mathbf{y}_k^i - \beta_k^i \mathbf{e}_i) \le f(\mathbf{y}_k^i - \overline{\beta_k^i}\mathbf{e}_i) - \gamma(\beta_k^i - \overline{\beta_k^i})^2 \le f(\mathbf{y}_k^i - \overline{\beta_k^i}\mathbf{e}_i).$$

By the mean value theorem, we have

$$f(\mathbf{y}_k^i - \beta_k^i \mathbf{e}_i) - f(\mathbf{y}_k^i - \overline{\beta_k^i}\mathbf{e}_i) = -(\beta_{k+1}^i - \overline{\beta_k^i})\nabla_i f(\boldsymbol{\xi}_k^i),$$

where $\boldsymbol{\xi}_k^i = \mathbf{y}_k^i - \overline{\beta_k^i}\mathbf{e}_i + t_k^i(\overline{\beta_k^i} - \beta_{k+1}^i)\mathbf{e}_i$ and $t_k^i \in (0,1)$. Then,

$$-\nabla_i f(\boldsymbol{\xi}_k^i) \le 0.$$

Hence, by similar reasoning as in case (1a), we obtain

$$(-\nabla f(\mathbf{x}_k))^\top(-\mathbf{e}_i) \le \left(L + L\sqrt{n}\right)\Delta_{k+1}. \tag{2.43}$$

(2a) $-\mathbf{e}_i \in G_k$, $(\mathbf{y}_k^{i+1})_i = u_i$ and $(\mathbf{y}_k^i)_i = u_i$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. Reasoning as in case (1a), with minor differences, we get

$$(-\nabla f(\mathbf{x}_k))^\top(-\mathbf{e}_i) \le \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1}. \tag{2.44}$$

(2b) $\mathbf{e}_i \in G_k$, $(\mathbf{y}_k^{i+1})_i = u_i$ and $(\mathbf{y}_k^i)_i < u_i$. Then, $\tilde{\alpha}_{k+1}^i = \alpha_k^i > 0$. Reasoning as in case (1b), with minor differences, we get

$$(-\nabla f(\mathbf{x}_k))^\top \mathbf{e}_i \le \left(L + L\sqrt{n}\right)\Delta_{k+1}. \tag{2.45}$$

(3a) $\{\pm\mathbf{e}_i\} \cap G_k \neq \emptyset$, $l_i < (\mathbf{y}_k^{i+1})_i < u_i$ and $(\mathbf{y}_k^i)_i = (\mathbf{y}_k^{i+1})_i$. Then, $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \nu_k^i$. From the instructions of the line-search procedure, we have that

$$\begin{aligned} f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) &> f(\mathbf{y}_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2 \quad \text{if } \mathbf{e}_i \in G_k, \\ f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) &> f(\mathbf{y}_k^i) - \gamma(\tilde{\alpha}_{k+1}^i)^2 \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned}$$

By the mean value theorem, we have

$$\begin{aligned} f(\mathbf{y}_k^i + \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - f(\mathbf{y}_k^i) &= \nabla_i f(\boldsymbol{\xi}_k^i)\tilde{\alpha}_{k+1}^i \quad \text{if } \mathbf{e}_i \in G_k, \\ f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - f(\mathbf{y}_k^i) &= -\nabla_i f(\bar{\boldsymbol{\xi}}_k^i)\tilde{\alpha}_{k+1}^i \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned}$$

where $\boldsymbol{\xi}_k^i = \mathbf{y}_k^i + t_k^i \tilde{\alpha}_{k+1}^i \mathbf{e}_i$ and $\bar{\boldsymbol{\xi}}_k^i = \mathbf{y}_k^i - \bar{t}_k^i \tilde{\alpha}_{k+1}^i \mathbf{e}_i$, with $t_k^i, \bar{t}_k^i \in (0,1)$. Then,

$$\begin{aligned} -\nabla_i f(\boldsymbol{\xi}_k^i) &< \gamma \tilde{\alpha}_{k+1}^i \quad \text{if } \mathbf{e}_i \in G_k, \\ \nabla_i f(\bar{\boldsymbol{\xi}}_k^i) &< \gamma \tilde{\alpha}_{k+1}^i \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned}$$

and, recalling that $t_k^i, \bar{t}_k^i \in (0,1)$, we also have

$$\begin{aligned} \|\mathbf{y}_k^i - \boldsymbol{\xi}_k^i\| &= t_k^i \tilde{\alpha}_{k+1}^i \le \tilde{\alpha}_{k+1}^i \quad \text{if } \mathbf{e}_i \in G_k, \\ \|\mathbf{y}_k^i - \bar{\boldsymbol{\xi}}_k^i\| &= t_k^i \tilde{\alpha}_{k+1}^i \le \tilde{\alpha}_{k+1}^i \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned}$$

Hence, reasoning as in case (1a) (i.e., using (2.41) for e_i and applying minor changes to (2.41), with $\boldsymbol{\xi}_k^i$ replaced by $\bar{\boldsymbol{\xi}}_k^i$, for $-\mathbf{e}_i$), we obtain

$$\begin{aligned} -\nabla f(\mathbf{x}_k)^\top \mathbf{e}_i &\le \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1} \quad \text{if } \mathbf{e}_i \in G_k, \\ -\nabla f(\mathbf{x}_k)^\top (-\mathbf{e}_i) &\le \left(\gamma + L + L\sqrt{n}\right)\Delta_{k+1} \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned} \tag{2.46}$$

(3b) $\mathbf{d}_k^i \in G_k$, $l_i < (\mathbf{y}_k^{i+1})_i < u_i$ and $(\mathbf{y}_k^i)_i \neq (\mathbf{y}_k^{i+1})_i$. Then, $\alpha_k^i > 0$, $\tilde{\alpha}_{k+1}^i = \alpha_k^i$. Let us assume that $\mathbf{d}_k^i = -\mathbf{e}_i$, i.e., $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i$ (the proof for the case $\mathbf{d}_k^i = \mathbf{e}_i$ is identical, except for minor changes). Then, from the instructions of the line-search procedure, there exist β_k^i and $\bar{\beta}_k^i$ such that $\beta_k^i \in \{0, \delta\tilde{\alpha}_{k+1}^i\}$, $\tilde{\alpha}_{k+1}^i < \bar{\beta}_k^i \le \tilde{\alpha}_{k+1}^i/\delta$ and

$$\begin{aligned} f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) &\le f(\mathbf{y}_k^i - \beta_k^i \mathbf{e}_i) - \gamma(\tilde{\alpha}_{k+1}^i - \beta_k^i)^2 \le f(\mathbf{y}_k^i - \beta_k^i \mathbf{e}_i), \\ f\left(\mathbf{y}_k^i - \bar{\beta}_k^i \mathbf{e}_i\right) &> f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - \gamma\left(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i\right)^2. \end{aligned}$$

By the mean value theorem, we have

$$\begin{aligned} f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) - f(\mathbf{y}_k^i - \beta_k^i \mathbf{e}_i) &= -(\tilde{\alpha}_{k+1}^i - \beta_k^i)\nabla_i f(\boldsymbol{\xi}_k^i), \\ f\left(\mathbf{y}_k^i - \bar{\beta}_k^i \mathbf{e}_i\right) - f(\mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i) &= -(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i)\nabla_i f(\bar{\boldsymbol{\xi}}_k^i), \end{aligned}$$

where $\boldsymbol{\xi}_k^i = \mathbf{y}_k^i - \beta_k^i \mathbf{e}_i - t_k^i(\tilde{\alpha}_{k+1}^i - \beta_k^i)\mathbf{e}_i$ and $\bar{\boldsymbol{\xi}}_k^i = \mathbf{y}_k^i - \tilde{\alpha}_{k+1}^i \mathbf{e}_i - \bar{t}_k^i(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i)\mathbf{e}_i$, with $t_k^i, \bar{t}_k^i \in (0,1)$. Then,

$$\begin{aligned} -\nabla_i f(\boldsymbol{\xi}_k^i) &\le 0, \\ \nabla_i f(\bar{\boldsymbol{\xi}}_k^i) &< \gamma(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i). \end{aligned}$$

Moreover, from the definition of $\bar{\beta}_k^i$, it follows that $0 \le \bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i \le (1/\delta - 1)\tilde{\alpha}_{k+1}^i \le \tilde{\alpha}_{k+1}^i/\delta$, where we have used the fact that $\delta \in (0,1)$. Then,

$$\begin{aligned} -\nabla_i f(\boldsymbol{\xi}_k^i) &\le 0, \\ \nabla_i f(\bar{\boldsymbol{\xi}}_k^i) &< \gamma \left(\frac{1-\delta}{\delta}\right) \tilde{\alpha}_{k+1}^i \le \gamma \frac{\tilde{\alpha}_{k+1}^i}{\delta}, \end{aligned}$$

and, recalling that $t_k^i, \bar{t}_k^i \in (0,1)$, we also have

$$\begin{aligned} \|\mathbf{y}_k^i - \boldsymbol{\xi}_k^i\| &= \beta_k^i + t_k^i(\tilde{\alpha}_{k+1}^i - \beta_k^i) \le \tilde{\alpha}_{k+1}^i \le \frac{\tilde{\alpha}_{k+1}^i}{\delta}, \\ \|\mathbf{y}_k^i - \bar{\boldsymbol{\xi}}_k^i\| &= \tilde{\alpha}_{k+1}^i + t_k^i(\bar{\beta}_k^i - \tilde{\alpha}_{k+1}^i) \le \bar{\beta}_k^i \le \frac{\tilde{\alpha}_{k+1}^i}{\delta}. \end{aligned}$$

Hence, reasoning as in the previous case, we obtain

$$\begin{aligned} -\nabla f(\mathbf{x}_k)^\top(-\mathbf{e}_i) = \nabla_i f(\mathbf{x}_k) &\le \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1}, \\ -\nabla f(\mathbf{x}_k)^\top \mathbf{e}_i = -\nabla_i f(\mathbf{x}_k) &\le \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1}. \end{aligned} \tag{2.47}$$

Hence, from (2.42), (2.43), (2.44), (2.45), (2.46) and (2.47), we conclude that

$$-\nabla f(\mathbf{x}_k)^\top \mathbf{d} \le \left(\frac{\gamma + L}{\delta} + L\sqrt{n}\right)\Delta_{k+1} \quad \forall \mathbf{d} \in G_k.$$

Now, let us analyze an iteration k such that $\mathbf{x}_{k+1} = \mathbf{x}_k$ (i.e., an unsuccessful iteration). In such a case, only cases (1a), (2a) and (3a) can occur, although we have to consider that $\mathbf{y}_k^i = \mathbf{x}_k$ and $\tilde{\alpha}_{k+1}^i = \theta \nu_k^i$. Hence, replacing L with L_i, we have that the following relations hold:

$$\begin{aligned} -\nabla f(\mathbf{x}_k)^\top \mathbf{e}_i &\le \frac{\gamma + L^{\max}}{\theta}\Delta_{k+1} \quad \text{if } \mathbf{e}_i \in G_k, \\ -\nabla f(\mathbf{x}_k)^\top(-\mathbf{e}_i) &\le \frac{\gamma + L^{\max}}{\theta}\Delta_{k+1} \quad \text{if } -\mathbf{e}_i \in G_k. \end{aligned}$$

As above, we finally conclude

$$-\nabla f(\mathbf{x}_k)^\top \mathbf{d} \le \frac{\gamma + L^{\max}}{\theta}\Delta_{k+1} \quad \forall \mathbf{d} \in G_k.$$

□

Remark 2.4.8. *The result expressed in Theorem 2.4.7 strongly relies on the extrapolation phase of the line-search procedure (i.e., lines 12–15 of* `Bounded DF-Linesearch`*). In particular, since we quit the expansion with a failure in the objective decrease when we do not hit the boundary of the feasible set, then we are able to upper bound* $\chi(\mathbf{x}_k)$ *for all iterations, including the successful ones. This represents a relevant difference over direct search [51] methods, where* $\chi(\mathbf{x}_k)$ *is usually upper bounded only for the unsuccessful iterations. Moreover, this property will allow us to give a complexity bound on the total number of iterations where* $\chi(\mathbf{x}_k)$ *is above a specified threshold (see Theorem 2.4.13 below).*

Combining Proposition 2.4.5 and Theorem 2.4.7, it is now straightforward to relate $\chi(\mathbf{x}_k)$ with Δ_{k+1}, as stated in the following result.

Theorem 2.4.9. *Let* $\{\mathbf{x}_k\}$ *be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Then,*

$$\chi(\mathbf{x}_k) \leq \begin{cases} \sqrt{n}\left(\dfrac{\gamma + L}{\delta} + L\sqrt{n} + \dfrac{M_g}{\theta}\right)\Delta_{k+1} & \text{if } \mathbf{x}_{k+1} \neq \mathbf{x}_k, \\ \sqrt{n}\left(\dfrac{\gamma + L^{max} + M_g}{\theta}\right)\Delta_{k+1} & \text{if } \mathbf{x}_{k+1} = \mathbf{x}_k. \end{cases}$$

Proof. Using Proposition 2.4.4 with $\epsilon = \Delta_k$ and Lemma 2.4.1, we obtain

$$\chi(\mathbf{x}_k) \leq \|(-\nabla f(\mathbf{x}_k))_{T_k}\| + \frac{\Delta_{k+1}}{\theta}\sqrt{n}M_g,$$

where we have used the fact that $\|(-\nabla f(\mathbf{x}_k))_{N(x_k,\epsilon)}\| \leq \|\nabla f(\mathbf{x}_k)\| \leq M_g$, where the last inequality follows from Assumption 2.2. So, using Proposition 2.4.5, it follows that there exists a direction $\mathbf{d} \in G_k$ such that

$$\chi(\mathbf{x}_k) \leq \sqrt{n}\left(-\nabla f(\mathbf{x}_k)^\top \mathbf{d} + \frac{M_g}{\theta}\Delta_{k+1}\right).$$

The desired result hence follows from Theorem 2.4.7. ☐

In order to get convergence to stationary points and provide worst-case complexity bounds for the proposed algorithm, for each iteration k let us define

$$\Phi_k = f(\mathbf{x}_k) + \eta\Delta_k^2, \tag{2.48}$$

where η satisfies

$$0 < \eta < \gamma(\delta(1-\delta))^2. \tag{2.49}$$

Recalling Assumption 2.1, note that

$$\Phi_k \geq f_{\min} \quad \forall k \geq 0. \tag{2.50}$$

Now, in the next theorem, we bound the difference $\Phi_k - \Phi_{k-1}$ for each iteration k. Note that the proof of the following theorem is equivalent to the one of Proposition 2.3.3 in the unconstrained case for the first and the third cases, while it differs from it for the second one. Indeed, in the bounded case, when we increase Δ_k, the last success in the `Bounded DF-Linesearch` might happen reaching the bound of the current coordinate, providing no precise information about the magnitude of the last expansion.

Theorem 2.4.10. *Let* $\{\mathbf{x}_k\}$ *be the sequence produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Then,*

$$\Phi_k - \Phi_{k-1} \leq -c_1\Delta_k^2 \quad \forall k \geq 1, \tag{2.51}$$

with

$$c_1 = \min\left\{\gamma c^2, \left(\gamma(\delta(1-\delta))^2 - \eta\right), \eta\left(\frac{1-\theta^2}{\theta^2}\right)\right\} > 0, \tag{2.52}$$

with η *satisfying* (2.49)*, and* c *and* θ *are defined in* `LAM`*.*

Proof. For each $k \geq 1$, consider the iteration $k-1$. The following cases can occur.

- $\mathbf{x}_k \neq \mathbf{x}_{k-1}$ (i.e., $k-1$ is a successful iteration) and $\Delta_k = \Delta_{k-1}$. Hence, there exists a coordinate $\bar{\imath}$ such that we have moved from $\mathbf{x}_{k-1}$ along $\pm e_{\bar{\imath}}$ and we have

  $$\alpha_{k-1}^{\bar{\imath}} \geq \nu_{k-1}^{\bar{\imath}} = \max\left\{\tilde{\alpha}_{k-1}^{\bar{\imath}}, c\Delta_{k-1}\right\} \geq c\Delta_{k-1} = c\Delta_k.$$

  Then, from the line-search procedure, we can write

  $$f(\mathbf{x}_k) \leq f(\mathbf{x}_{k-1}) - \gamma c^2 \Delta_k^2.$$

  Hence, we have

  $$\Phi_k - \Phi_{k-1} = f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right) \leq -\gamma c^2 \Delta_k^2. \tag{2.53}$$

- $\mathbf{x}_k \neq \mathbf{x}_{k-1}$ (i.e., $k-1$ is a successful iteration) and $\Delta_k > \Delta_{k-1}$.
  Hence, there exists a coordinate $\bar{\imath}$ such that

  $$\alpha_{k-1}^{\bar{\imath}} = \Delta_k.$$

  Let h_{k-1}^i be the number of times the step-size related to the i-th coordinate is expanded on iteration $k-1$, that is, if $\alpha_k^i > 0$

  $$\alpha_{k-1}^i = \min\{\alpha_{\mathtt{max}}, \delta^{-h_{k-1}^i}\nu_{k-1}^i\},$$

  where $\alpha_{\mathtt{max}}$ is the distance to the bound of the i-th coordinate over $\mathbf{d}_{k-1}^i$. Now, considering $\bar{\imath}$, we have

  $$\alpha_{k-1}^{\bar{\imath}} > \Delta_{k-1} \geq \nu_{k-1}^{\bar{\imath}},$$

  which implies that $h_{k-1}^{\bar{\imath}} \geq 1$, otherwise we would have $\alpha_{k-1}^{\bar{\imath}} \in \{0, \nu_{k-1}^{\bar{\imath}}\}$.
  If $h_{k-1}^{\bar{\imath}} = 1$, then we have

  $$\Delta_k = \alpha_{k-1}^{\bar{\imath}} = \min\{\alpha_{\max}, \delta^{-1}\nu_{k-1}^{\bar{\imath}}\} \leq \delta^{-1}\nu_{k-1}^{\bar{\imath}} \implies \nu_{k-1}^{\bar{\imath}} \geq \delta\Delta_k$$

  Thus, we get

  $$f(\mathbf{x}_k) \leq f(\mathbf{x}_{k-1}) - \gamma(\nu_{k-1}^{\bar{\imath}})^2 \leq f(\mathbf{x}_{k-1}) - \gamma\delta^2\Delta_k^2. \tag{2.54}$$

  Let us now consider $h_{k-1}^{\bar{\imath}} \geq 2$. We have

  $$\Delta_k = \alpha_{k-1}^{\bar{\imath}} = \min\{\alpha_{\max}, \delta^{-h_{k-1}^{\bar{\imath}}}\nu_{k-1}^{\bar{\imath}}\} \leq \delta^{-h_{k-1}^{\bar{\imath}}}\nu_{k-1}^{\bar{\imath}}$$

  $$\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}} \geq \delta\Delta_k. \tag{2.55}$$

  Therefore we can write

  $$\begin{aligned} f(\mathbf{x}_k) &\leq f(\mathbf{x}_{k-1}) - \gamma\left(\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}} - \delta^{-(h_{k-1}^{\bar{\imath}}-2)}\nu_{k-1}^{\bar{\imath}}\right)^2 \\ &= f(\mathbf{x}_{k-1}) - \gamma\left(\delta^{-(h_{k-1}^{\bar{\imath}}-1)}\nu_{k-1}^{\bar{\imath}}(1-\delta)\right)^2 \\ \text{using (2.55)} \quad &\leq f(\mathbf{x}_{k-1}) - \gamma\left(\delta(1-\delta)\right)^2\Delta_k^2. \end{aligned} \tag{2.56}$$

  Finally, using (2.54) and (2.56), and considering that $\min\{(1-\delta)^2, (\delta(1-\delta))^2\} = (\delta(1-\delta))^2$, we can write

  $$f(\mathbf{x}_k) \leq f(\mathbf{x}_{k-1}) - \gamma(\delta(1-\delta))^2\Delta_k^2.$$

Hence, we have

$$\begin{aligned}\Phi_k - \Phi_{k-1} &= f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right)\\ &\le -\gamma(\delta(1-\delta))^2\Delta_k^2 + \eta\Delta_k^2\\ &= -\left(\gamma(\delta(1-\delta))^2 - \eta\right)\Delta_k^2.\end{aligned} \tag{2.57}$$

- $\mathbf{x}_k = \mathbf{x}_{k-1}$ (i.e., $k-1$ is an unsuccessful iteration). Recalling Lemma 2.4.1, we have

$$\Delta_k = \theta\Delta_{k-1}.$$

Then,

$$\begin{aligned}\Phi_k - \Phi_{k-1} &= f(\mathbf{x}_k) - f(\mathbf{x}_{k-1}) + \eta\left(\Delta_k^2 - \Delta_{k-1}^2\right)\\ &= \eta\left(\Delta_k^2 - \frac{1}{\theta^2}\Delta_k^2\right)\\ &= -\eta\left(\frac{1-\theta^2}{\theta^2}\right)\Delta_k^2.\end{aligned} \tag{2.58}$$

Finally, we get (2.51) by combining (2.53), (2.57) and (2.58), where $c_1 > 0$ since $c \in (0,1]$. ☐

Using the above result, we can easily show the convergence to zero of the sequences of tentative and actual step-sizes produced by the algorithm, i.e., $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$, $i = 1,\dots,n$, respectively, together with the convergence to zero of the sequence of maximum tentantive step-sizes $\{\Delta_k\}$.

Proposition 2.4.11. *Let $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$, $i = 1,\dots,n$, and $\{\Delta_k\}$ be the sequences produced by* `LAM` *using the* `Bounded DF-Linesearch`*. We have that*

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0, \quad i = 1,\dots,n; \tag{2.59}$$

$$\lim_{k\to\infty} \alpha_k^i = 0, \quad i = 1,\dots,n; \tag{2.60}$$

$$\lim_{k\to\infty} \Delta_k = 0. \tag{2.61}$$

Proof. From (2.50) and (2.51), we get (2.61). Then, using the definition of Δ_k given in *LAM*, also (2.59) holds. Since, for all $k \ge 1$ and for all $i = 1,\dots,n$, from the instructions of the algorithm either $\alpha_{k-1}^i = 0$ or $\alpha_{k-1}^i = \tilde{\alpha}_k^i$, then (2.60) follows from (2.59). ☐

Using Theorem 2.4.9 and Proposition 2.4.11, it is now possible to prove the convergence of the algorithm to stationary points.

Theorem 2.4.12. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Then,*

- $\lim_{k\to\infty} \chi(\mathbf{x}_k) = 0$, *i.e., every limit point of $\{\mathbf{x}_k\}$ is stationary,*
- $\lim_{k\to\infty} \|\mathbf{x}_{k+1} - \mathbf{x}_k\| = 0.$

Proof. From Theorem 2.4.9 and (2.61) in Proposition 2.4.11, taking the limit for $k \to \infty$ it follows that $\chi(\mathbf{x}_k) \to 0$, that is, every limit point of $\{\mathbf{x}_k\}$ is stationary. Finally, since $\mathbf{x}_{k+1} - \mathbf{x}_k = \sum_{i=1}^n \alpha_k^i \mathbf{d}_k^i$, using (2.60) in Proposition 2.4.11 we also have that $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \to 0$. ☐

2.4.4 Worst-case complexity

This section is devoted to analyze the worst-case complexity of `LAM` using the `Bounded DF-Linesearch`. In particular,

(i) we give an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$ on the total number of iterations where $\chi(\mathbf{x}_k)$ is not below a specified threshold ϵ;

(ii) we give an upper bound of $\mathcal{O}(n\epsilon^{-2})$ on the number of iterations required to generate the first point $\mathbf{x}_k$ where $\chi(\mathbf{x}_k)$ is below a specified tolerance ϵ;

(iii) we give an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$ on the number of function evaluations required to generate the first point $\mathbf{x}_k$ where $\chi(\mathbf{x}_k)$ is below a specified tolerance ϵ.

We start by providing an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$ on the total number of iterations where $\chi(\mathbf{x}_k) \geq \epsilon$, with a given $\epsilon > 0$.

Theorem 2.4.13. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Given any $\epsilon > 0$, let*

$$\mathcal{K}_\epsilon = \{k\colon \chi(\mathbf{x}_k) \geq \epsilon\}.$$

Then, $|\mathcal{K}_\epsilon| \leq \mathcal{O}(n^2\epsilon^{-2})$. In particular,

$$|\mathcal{K}_\epsilon| \leq \left\lfloor \frac{c_2^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor,$$

where c_1 is defined as in Theorem 2.4.10 and

$$c_2 = \sqrt{n}\max\left\{\frac{\gamma + L}{\delta} + L\sqrt{n} + \frac{M_g}{\theta}, \frac{\gamma + L^{max} + M_g}{\theta}\right\}.$$

Proof. From Theorem 2.4.10, it follows that the sequence $\{\Phi_k\}$ is monotonically non-increasing. Furthermore, for all $k \geq 1$, we can write

$$\Phi_k - \Phi_0 \leq -c_1 \sum_{j=1}^{k} \Delta_j^2 = -c_1 \sum_{j=0}^{k-1} \Delta_{j+1}^2. \tag{2.62}$$

Since the sequence $\{\Phi_k\}$ is bounded from below, there exists Φ^* such that

$$\lim_{k\to\infty} \Phi_k = \Phi^* \geq f_{\min},$$

with $f_{\min}$ defined as in (2.50). Taking the limit for $k \to \infty$ in (2.62) we obtain

$$\Phi_0 - f_{\min} \geq \Phi_0 - \Phi^* \geq c_1 \sum_{k=0}^{\infty} \Delta_{k+1}^2 \geq c_1 \sum_{k\in\mathcal{K}_\epsilon} \Delta_{k+1}^2.$$

Therefore, using the definition of K_ϵ and Theorem 2.4.9, we get

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k\in\mathcal{K}_\epsilon} \Delta_{k+1}^2 \geq |\mathcal{K}_\epsilon| c_1 \frac{\epsilon^2}{c_2^2}.$$

Thus, the desired result is obtained. □

As appears from the proof of Theorem 2.4.13, the above result relies on Theorem 2.4.9 which, in turn, uses Theorem 2.40. The latter, as pointed out in Remark 2.4.8, strongly relies on the extrapolation phase of the line-search procedure (i.e., lines 12–15 of the `Bounded DF-Linesearch`), which allows us to bound $\chi(\mathbf{x}_k)$ at both successful and unsuccessful iterations.

In the following theorem, we give an upper bound of $\mathcal{O}(n\epsilon^{-2})$ on the maximum number of iterations required to produce a point $\mathbf{x}_k$ such that $\chi(\mathbf{x}_k)$ is below a given threshold $\epsilon > 0$. This bound aligns with established findings for direct search [51] and model-based [59] methods.

Theorem 2.4.14. *Let* $\{\mathbf{x}_k\}$ *be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Given any* $\epsilon > 0$*, let* $j_\epsilon \geq 1$ *be the first iteration such that* $\chi(\mathbf{x}_{j_\epsilon}) < \epsilon$*, that is,* $\chi(\mathbf{x}_k) \geq \epsilon$ *for all* $k \in \{0, \dots, j_\epsilon - 1\}$*.*

Then, $j_\epsilon \leq \mathcal{O}(n\epsilon^{-2})$*. In particular,*

$$j_\epsilon \leq \left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor,$$

where c_1 *is given by* (2.52) *and*

$$c_3 = \frac{\gamma + L^{max} + M_g}{\theta}. \tag{2.63}$$

Proof. Let Φ_k the function defined in (2.48). We can write

$$\Phi_{j_\epsilon} - \Phi_0 = \sum_{k=0}^{j_\epsilon - 1} (\Phi_{k+1} - \Phi_k)$$

and, using Theorem 2.4.10, we have that

$$\Phi_{j_\epsilon} - \Phi_0 \leq -c_1 \sum_{k=0}^{j_\epsilon - 1} \Delta_{k+1}^2.$$

Recalling (2.50) and the fact that $\Phi_k \geq f(\mathbf{x}_k)$ for $k \geq 0$, we get

$$f_{\min} - \Phi_0 \leq \Phi_{j_\epsilon} - \Phi_0 \leq -c_1 \sum_{k=0}^{j_\epsilon - 1} \Delta_{k+1}^2. \tag{2.64}$$

Now, we can partition the set of iteration indices $\{0, \dots, j_\epsilon - 1\}$ into $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ such that

$$k \in \mathcal{S}_{j_\epsilon} \Leftrightarrow \mathbf{x}_k \neq \mathbf{x}_{k-1}, \quad k \in \mathcal{U}_{j_\epsilon} \Leftrightarrow \mathbf{x}_k = \mathbf{x}_{k-1}, \quad \mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0, \dots, j_\epsilon - 1\},$$

that is, $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ contain the successful and unsuccessful iterations up to $j_\epsilon - 1$, respectively. So, from (2.64), we can write

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k \in \mathcal{S}_{j_\epsilon}} \Delta_{k+1}^2 + c_1 \sum_{k \in \mathcal{U}_{j_\epsilon}} \Delta_{k+1}^2. \tag{2.65}$$

For all $k \in \mathcal{S}_{j_\epsilon}$, let us define the index $m(k)$ as follows:

- if $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\} \neq \emptyset$, then $m(k)$ is the largest index of $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\}$;
- otherwise, $m(k) = -1$.

Note that, by definition, $m(k)$ is the last unsuccessful iteration before iteration k, i.e., all the iterations from $m(k)+1$ to k are successful iterations. Lemma 2.4.1 guarantees that $\Delta_{k+1} \geq \Delta_{m(k)+1}$ for all $k \in \mathcal{S}_{j_\epsilon}$. Using (2.65), we obtain

$$\Phi_0 - f_{\min} \geq c_1 \sum_{k\in\mathcal{S}_{j_\epsilon}} \Delta^2_{m(k)+1} + c_1 \sum_{k\in\mathcal{U}_{j_\epsilon}} \Delta^2_{k+1}.$$

From Theorem 2.4.9, we have that $\Delta_{m(k)+1} \geq \chi(\mathbf{x}_{m(k)})/(\sqrt{n}c_3)$ for all $k \in \mathcal{S}_{j_\epsilon}$ and $\Delta_{k+1} \geq \chi(\mathbf{x}_k)/(\sqrt{n}c_2)$ for all $k \in \mathcal{U}_{j_\epsilon}$. Since $\chi(\mathbf{x}_k) \geq \epsilon$ for all $k \in \{0,\dots,j_\epsilon-1\}$, with $\mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0,\dots,j_\epsilon-1\}$, we get

$$\Phi_0 - f_{\min} \geq j_\epsilon \frac{c_1}{nc_3^2}\epsilon^2.$$

Thus, the desired result is obtained. □

The last complexity result we give is about the maximum number of function evaluations required to produce a point $\mathbf{x}_k$ such that $\chi(\mathbf{x}_k)$ is less than or equal to a given threshold $\epsilon > 0$. Using arguments from the related literature [86] and Section 2.3, we obtain an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$, which still aligns with established findings for direct search [51] and model-based [59] methods

Theorem 2.4.15. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch`*. Given any $\epsilon > 0$, let $j_\epsilon \geq 1$ be the first iteration such that $\chi(\mathbf{x}_{j_\epsilon}) < \epsilon$, that is, $\chi(\mathbf{x}_k) \geq \epsilon$ for all $k \in \{0,\dots,j_\epsilon - 1\}$.*

Denoting by Nf_{j_ϵ} the number of function evaluations required by `LAM` *using the* `Bounded DF-Linesearch` *up to iteration j_ϵ, then $Nf_{j_\epsilon} \leq \mathcal{O}(n^2\epsilon^{-2})$. In particular,*

$$Nf_{j_\epsilon} \leq 2n \left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor + \left\lfloor \frac{nc_3^2(f_0 - f_{\min})}{\gamma c^2} \max\left\{1, \left(\frac{\delta}{1-\delta}\right)^2\right\}\epsilon^{-2} \right\rfloor,$$

where c_1 and c_3 are given in (2.52) *and* (2.63)*, respectively.*

Proof. First, let us partition the set of iteration indices $\{0,\dots,j_\epsilon - 1\}$ into $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ such that

$$k \in \mathcal{S}_{j_\epsilon} \Leftrightarrow \mathbf{x}_k \neq \mathbf{x}_{k-1}, \quad k \in \mathcal{U}_{j_\epsilon} \Leftrightarrow \mathbf{x}_k = \mathbf{x}_{k-1}, \quad \mathcal{S}_{j_\epsilon} \cup \mathcal{U}_{j_\epsilon} = \{0,\dots,j_\epsilon - 1\},$$

that is, $\mathcal{S}_{j_\epsilon}$ and $\mathcal{U}_{j_\epsilon}$ contain the successful and unsuccessful iterations up to $j_\epsilon - 1$, respectively.

When the algorithm evaluates a new point, the latter can either succeed to decrease the objective function or fail to do so. Note that we consider the function evaluations where the extrapolation has led to the variables bounds as failing ones, since we cannot be sure if the performed step length is somehow related to Δ_k. Let us then define $\#f^{\mathcal{S}}_{j_\epsilon}$ as the total number of function evaluations related to points which succeed to decrease the objective function up to iteration j_ϵ. Note that, for each iteration, the maximum number of function evaluations related to points which fail to decrease the objective function is $2n$ (and it can be equal to $2n$ only when $T_k = \mathbb{R}^n$). So, we can write

$$Nf_{j_\epsilon} \leq 2nj_\epsilon + \#f^{\mathcal{S}}_{j_\epsilon} \leq 2n\left\lfloor \frac{nc_3^2(\Phi_0 - f_{\min})}{c_1}\epsilon^{-2} \right\rfloor + \#f^{\mathcal{S}}_{j_\epsilon}, \tag{2.66}$$

where the last inequality follows from Theorem 2.4.14. Now, let us consider any iteration $k < j_\epsilon$ and any index $i \in \{1,\dots,n\}$ such that the line-search succeeds to produce a decrease in the objective function. For each α used in the extrapolation phase of the line-search, we have that either

$$f(\mathbf{y}^i_k) - f(\mathbf{y}^i_k + \alpha\mathbf{d}^i_k) \geq \gamma\alpha^2 \geq \gamma c^2\Delta^2_k, \tag{2.67}$$

or

$$f(\mathbf{y}_k^i + \alpha \mathbf{d}_k^i) - f(\mathbf{y}_k^i + (\alpha/\delta)\mathbf{d}_k^i) \geq \gamma \left(\frac{1-\delta}{\delta} \right)^2 \alpha^2 \geq \gamma \left(\frac{1-\delta}{\delta} \right)^2 c^2 \Delta_k^2. \tag{2.68}$$

Let us define the index $m(k)$ as follows:

- if $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\} \neq \emptyset$, then $m(k)$ is the largest index of $\mathcal{U}_{j_\epsilon} \cap \{0, \dots, k-1\}$;
- otherwise, $m(k) = 0$.

Note that, by definition, $m(k)$ is the last unsuccessful iteration before iteration k, i.e., all the iterations from $m(k)+1$ to k are successful iterations. Lemma 2.4.1 guarantees that $\Delta_k \geq \Delta_{m(k)+1}$ for all $k \in \mathcal{S}_{j_\epsilon}$. Hence, from (2.67) and (2.68), it follows that

$$f(\mathbf{y}_k^i) - f(\mathbf{y}_k^i + \alpha \mathbf{d}_k^i) \geq \gamma \alpha^2 \geq \gamma c^2 \Delta_{m(k)+1}^2 \geq \gamma \min \left\{ 1, \left(\frac{1-\delta}{\delta} \right)^2 \right\} c^2 \Delta_{m(k)+1}^2,$$

and

$$f(\mathbf{y}_k^i + \alpha \mathbf{d}_k^i) - f(\mathbf{y}_k^i + (\alpha/\delta)\mathbf{d}_k^i) \geq \gamma \left(\frac{1-\delta}{\delta} \right)^2 \alpha^2 \geq \gamma \min \left\{ 1, \left(\frac{1-\delta}{\delta} \right)^2 \right\} c^2 \Delta_{m(k)+1}^2.$$

From Theorem 2.4.9, we have that $\Delta_{m(k)+1} \geq \chi(\mathbf{x}_{m(k)})/(\sqrt{n} c_3)$. Since $\chi(\mathbf{x}_k) \geq \epsilon$ for all $k \in \{0, \dots, j_\epsilon - 1\}$, we can write

$$f(\mathbf{y}_k^i) - f(\mathbf{y}_k^i + \alpha \mathbf{d}_k^i) \geq \gamma \min \left\{ 1, \left(\frac{1-\delta}{\delta} \right)^2 \right\} c^2 \frac{\epsilon^2}{n c_3^2},$$

and

$$f(\mathbf{y}_k^i + \alpha \mathbf{d}_k^i) - f(\mathbf{y}_k^i + (\alpha/\delta)\mathbf{d}_k^i) \geq \gamma \min \left\{ 1, \left(\frac{1-\delta}{\delta} \right)^2 \right\} c^2 \frac{\epsilon^2}{n c_3^2}.$$

Then, recalling Assumption 2.1 and summing up the above relation over all function evaluations producing an objective decrease, we obtain

$$f_0 - f_{\min} \geq \# f_{j_\epsilon}^{\mathcal{S}} \gamma \min \left\{ 1, \left(\frac{1-\delta}{\delta} \right)^2 \right\} c^2 \frac{\epsilon^2}{n c_3^2},$$

that is,

$$\# f_{j_\epsilon}^{\mathcal{S}} \leq \left\lfloor \frac{n c_3^2 (f_0 - f_{\min})}{\gamma c^2} \max \left\{ 1, \left(\frac{\delta}{1-\delta} \right)^2 \right\} \epsilon^{-2} \right\rfloor.$$

The desired results hence follows from (2.66). □

2.4.5 Finite active-set identification

In this section, we show that `LAM` using the `Bounded DF-Linesearch` identifies the components of the final solution lying on the lower or the upper bounds (the so called *active set*) in a finite number of iterations.

First, let us give an equivalent definition of stationarity for problem (P1) when $\Omega = [\mathbf{l}, \mathbf{u}]$, which will be useful for our analysis.

Definition 2.4.16. *A point* $\mathbf{x}^* \in [\mathbf{l}, \mathbf{u}]$ *is a stationary point of problem* (P1) *(i.e.,* $\chi(\mathbf{x}^*) = 0$*) if and only if, for all* $i \in \{1, \dots, n\}$*, we have that*

$$\nabla_i f(\mathbf{x}^*) \begin{cases} \geq 0 & \text{if } x*_i = l_i, \\ = 0 & \text{if } l_i < x_i^* < u_i, \\ \leq 0 & \text{if } x_i^* = u_i. \end{cases}$$

Now, let us recall the definition of *strict complementarity* and *non-degenerate solutions.*

Definition 2.4.17. *Given a stationary point* $\mathbf{x}^*$ *of problem* (P1)*, we say that a component* x_i^* *satisfies the strict complementarity condition if* $x_i^* \in \{l_i, u_i\}$ *and* $\nabla_i f(\mathbf{x}^*) \neq 0$*. If the strict complementarity condition is satisfied by all components* x_i^**, we say that* $\mathbf{x}^*$ *is non-degenerate.*

In particular, we define $\mathscr{Z}(\mathbf{x}^*)$ as the *active set* for a stationary point $\mathbf{x}^*$, that is, the index set for the active components of $\mathbf{x}^*$. We also define $\mathscr{Z}^+(\mathbf{x}^*)$ as the index set for those components satisfying the strict complementarity condition. Namely,

$$\mathscr{Z}(\mathbf{x}^*) = \{i \colon x_i^* = l_i\} \cup \{i \colon x_i^* = u_i\} \qquad \text{and} \qquad \mathscr{Z}^+(\mathbf{x}^*) = \mathscr{Z}(\mathbf{x}^*) \cap \{i \colon \nabla_i f(\mathbf{x}^*) \neq 0\}.$$

Furthermore, for any stationary point $\mathbf{x}^*$ such that $\mathscr{Z}^+(\mathbf{x}^*) \neq \emptyset$, let us define

$$\zeta(\mathbf{x}^*) = \min_{i \in \mathscr{Z}^+(\mathbf{x}^*)} |\nabla_i f(\mathbf{x}^*)|. \tag{2.69}$$

We see that $\zeta(\mathbf{x}^*)$ is a measure of the minimum amount of strict complementarity among the variable in $\mathscr{Z}^+(\mathbf{x}^*)$. This quantity will be used to define a neighborhood of $\mathbf{x}^*$ where the active components are correctly identified, following a similar approach as in [32, 81].

Before diving into the main theorem of this section, we need some preliminary results following from the Lipschitz continuity of ∇f. Recalling Definition 2.4.2, using standard arguments (see, e.g., [79]) one can prove that, for all $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$, we have

$$|f(\mathbf{x} + s\mathbf{e}_i) - f(\mathbf{x}) - s\nabla_i f(\mathbf{x})| \leq \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}], \quad i = 1, \dots, n.$$

Hence, for all $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$, we have

$$f(\mathbf{x} + s\mathbf{e}_i) \leq f(\mathbf{x}) + s\nabla_i f(\mathbf{x}) + \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}], \quad i = 1, \dots, n, \tag{2.70}$$

$$f(\mathbf{x} + s\mathbf{e}_i) \geq f(\mathbf{x}) + s\nabla_i f(\mathbf{x}) - \frac{L_i}{2} s^2 \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}], \quad i = 1, \dots, n. \tag{2.71}$$

The two following results provide bounds for the objective function when exploring any coordinate direction.

Proposition 2.4.18. *Given* $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$, $\gamma \geq 0$ *and* $i \in \{1, \dots, n\}$, *then*

$$f(\mathbf{x} - s \operatorname{sign}(\nabla_i f(\mathbf{x}))\mathbf{e}_i) \leq f(\mathbf{x}) - \gamma s^2$$

for all $0 \leq s \leq 2\dfrac{|\nabla_i f(\mathbf{x})|}{L_i + 2\gamma}$ *such that* $\mathbf{x} - s \operatorname{sign}(\nabla_i f(\mathbf{x}))\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}]$.

Proof. From (2.70), we can write

$$f(\mathbf{x} + s\mathbf{e}_i) \leq f(\mathbf{x}) + s\Big(\nabla_i f(\mathbf{x}) + \frac{L_i}{2}s\Big) \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}].$$

The right-hand side of the above inequality is less than or equal to $f(\mathbf{x}) - \gamma s^2$ if

$$s\Big(\nabla_i f(\mathbf{x}) + \frac{L_i}{2}s\Big) \leq -\gamma s^2.$$

If $\nabla_i f(\mathbf{x}) \neq 0$, solving with respect to s we obtain

$$\begin{aligned} -\frac{2\nabla_i f(\mathbf{x})}{L_i + 2\gamma} \leq s \leq 0 \quad & \text{if } \nabla_i f(\mathbf{x}) > 0, \\ 0 \leq s \leq -\frac{2\nabla_i f(\mathbf{x})}{L_i + 2\gamma} \quad & \text{if } \nabla_i f(\mathbf{x}) < 0, \end{aligned}$$

leading to the desired result. ☐

Proposition 2.4.19. *Given* $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$ *and* $i \in \{1, \dots, n\}$, *then*

$$f(\mathbf{x} + s \operatorname{sign}(\nabla_i f(\mathbf{x}))\mathbf{e}_i) \geq f(\mathbf{x})$$

for all $0 \leq s \leq \dfrac{2|\nabla_i f(\mathbf{x})|}{L_i}$ *such that* $\mathbf{x} + s \operatorname{sign}(\nabla_i f(\mathbf{x}))\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}]$.

Proof. From (2.71), we can write

$$f(\mathbf{x} + s\mathbf{e}_i) \geq f(\mathbf{x}) + s\Big(\nabla_i f(\mathbf{x}) - \frac{L_i}{2}s\Big) \quad \forall s \in \mathbb{R} \colon \mathbf{x} + s\mathbf{e}_i \in [\mathbf{l}, \mathbf{u}].$$

The right-hand side of the above inequality is greater than or equal to $f(\mathbf{x})$ if

$$s\Big(\nabla_i f(\mathbf{x}) - \frac{L_i}{2}s\Big) \geq 0.$$

If $\nabla_i f(\mathbf{x}) \neq 0$, solving with respect to s we obtain

$$\begin{aligned} 0 \leq s \leq \frac{2\nabla_i f(\mathbf{x})}{L_i} \quad & \text{if } \nabla_i f(\mathbf{x}) > 0, \\ \frac{2\nabla_i f(\mathbf{x})}{L_i} \leq s \leq 0 \quad & \text{if } \nabla_i f(\mathbf{x}) < 0, \end{aligned}$$

leading to the desired result. ☐

The next proposition shows that, when ν_k^i is sufficiently small at a given iteration, `LAM` using the `Bounded DF-Linesearch` cannot move along an ascent direction.

Proposition 2.4.20. *Consider an iteration k of* `LAM` *using the* `Bounded DF-Linesearch`*. If $\nu_k^i \le 2|\nabla_i f(\mathbf{y}_k^i)|/L_i$ for an index $i \in \{1,\dots,n\}$, then*

$$\alpha_k^i > 0 \;\Rightarrow\; \mathbf{d}_k^i = -\operatorname{sign}(\nabla_i f(\mathbf{y}_k^i))e_i.$$

Proof. Using Proposition 2.4.19, for all $\alpha \le \nu_k^i$ and $\gamma > 0$ we have

$$f(\mathbf{y}_k^i + \alpha \operatorname{sign}(\nabla_i f(\mathbf{y}_k^i))\mathbf{e}_i) \ge f(\mathbf{y}_k^i) > f(\mathbf{y}_k^i) - \gamma\alpha^2.$$

That is the line-search in the *Bounded DF-Linesearch* fails when using the direction $\operatorname{sign}(\nabla_i f(\mathbf{y}_k^i))\mathbf{e}_i$ with any step-size $0 < \alpha \le \nu_k^i$. So, if the line-search returns $\alpha_k^i > 0$, necessarily $\mathbf{d}_k^i = -\operatorname{sign}(\nabla_i f(\mathbf{y}_k^i))\mathbf{e}_i$. □

Now, we are ready to state the main result of this section, establishing finite active-set identification of Algorithm 6.

Theorem 2.4.21. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch` *and let $\mathbf{x}^*$ be a limit point of $\{\mathbf{x}_k\}$, i.e., there exists an infinite subsequence $\{\mathbf{x}_k\}_{\mathcal{K}^{\mathrm{x}}} \to \mathbf{x}^*$. Then,*

(i) $\displaystyle\lim_{k\to\infty,\, k\in\mathcal{K}^{\mathrm{x}}} \mathbf{x}_{k+1} = \mathbf{x}^*$;

(ii) an iteration $\bar{k} \in \mathcal{K}^{\mathrm{x}}$ exists such that, for all $k \ge \bar{k}$, $k \in \mathcal{K}^{\mathrm{x}}$, we have that $(\mathbf{x}_{k+1})_i = \mathbf{x}_i^*$ for all $i \in \mathscr{Z}^+(\mathbf{x}^*)$.

Proof. Since $\mathbf{y}_k^1 = \mathbf{x}_k$ and $\mathbf{y}_k^{i+1} = \mathbf{x}_k + \sum_{j=1}^i \alpha_k^j \mathbf{d}_k^j$, from Proposition 2.4.11 and the fact that $\|\mathbf{d}_k^i\| = 1$, $i = 1,\dots,n+1$, we have

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{K}^{\mathrm{x}}}} \mathbf{y}_k^i = \lim_{\substack{k\to\infty\\ k\in\mathcal{K}^{\mathrm{x}}}} \mathbf{x}_k = \mathbf{x}^*, \quad i = 1,\dots,n+1. \tag{2.72}$$

Since $\mathbf{x}_{k+1} = \mathbf{y}_k^{n+1}$, then point (i) follows.

To show point (ii), assume that $\mathscr{Z}^+(\mathbf{x}^*) \ne \emptyset$. Let $\bar{k} \in \mathcal{K}^{\mathrm{x}}$ be the first iteration such that the two following relations hold for all $k \ge \bar{k}$, $k \in \mathcal{K}^{\mathrm{x}}$:

$$\|\mathbf{y}_k^i - \mathbf{x}^*\| \le \min\left\{\frac{1}{L}, \frac{2}{2L + L^{\max} + 2\gamma}\right\}\zeta(\mathbf{x}^*), \quad i = 1,\dots,n, \tag{2.73a}$$

$$\|\mathbf{y}_k^i - \mathbf{x}^*\| + \frac{L^{\max}}{2L}\max_{j=1,\dots,n} \tilde{\alpha}_k^j \le \frac{\zeta(\mathbf{x}^*)}{L}, \quad i = 1,\dots,n. \tag{2.73b}$$

Note that (2.72) and Proposition 2.4.11 imply the existence of $\bar{k} \in \mathcal{K}^{\mathrm{x}}$ such that (2.73) holds for all $k \ge \bar{k}$, $k \in \mathcal{K}^{\mathrm{x}}$.

Consider an index $i \in \mathscr{Z}^+(\mathbf{x}^*)$ and an iteration $k \ge \bar{k}$, $k \in \mathcal{K}^{\mathrm{x}}$. To prove that $(\mathbf{x}_{k+1})_i = x_i^*$, we have to show that $(\mathbf{y}_k^{i+1})_i = x_i^*$ since, from the instructions of the algorithm, $(\mathbf{x}_{k+1})_i = (\mathbf{y}_k^{i+1})_i$. Without loss of generality, assume that $x_i^* = l_i$ (the proof for the case $x_i^* = u_i$ is identical, except for minor changes). So, we have to show that

$$(\mathbf{y}_k^{i+1})_i = l_i. \tag{2.74}$$

Preliminarily, we want to prove that

$$|z_i - l_i| \le \frac{2\nabla_i f(\mathbf{z})}{L_i + 2\gamma} \quad \forall \mathbf{z} \in \mathbb{R}^n \text{ such that } \|\mathbf{z} - \mathbf{x}^*\| \le \|\mathbf{y}_k^i - \mathbf{x}^*\|, \tag{2.75}$$

$$\nu_k^i \le \frac{2\nabla_i f(\mathbf{y}_k^i)}{L_i}. \tag{2.76}$$

According to Definitions 2.4.16 and 2.4.17, we have that $\nabla_i f(\mathbf{x}^*) > 0$ and, from the definition of $\zeta(\mathbf{x}^*)$ given in (2.69), it follows that

$$0 < \zeta(\mathbf{x}^*) \le \nabla_i f(\mathbf{x}^*). \tag{2.77}$$

Consider any $\mathbf{z} \in \mathbb{R}^n$ such that $\|\mathbf{z} - \mathbf{x}^*\| \le \|\mathbf{y}_k^i - \mathbf{x}^*\|$. Using the Lipschitz continuity of ∇f, we have

$$\nabla_i f(\mathbf{x}^*) - \nabla_i f(\mathbf{z}) \le \|\nabla f(\mathbf{z}) - \nabla f(\mathbf{x}^*)\| \le L\|\mathbf{z} - \mathbf{x}^*\|. \tag{2.78}$$

Moreover, from (2.73a) we can write

$$\|\mathbf{z} - \mathbf{x}^*\| \le \|\mathbf{y}_k^i - \mathbf{x}^*\| \le \frac{2\zeta(\mathbf{x}^*)}{2L + L^{\max} + 2\gamma}.$$

Multiplying the first and last terms above by $(2L + L^{\max} + 2\gamma)/(L^{\max} + 2\gamma)$, we have

$$\Big(\frac{2L}{L^{\max} + 2\gamma} + 1\Big)\|\mathbf{z} - \mathbf{x}^*\| \le \frac{2\zeta(\mathbf{x}^*)}{L^{\max} + 2\gamma},$$

that is,

$$\|\mathbf{z} - \mathbf{x}^*\| \le \big(\zeta(\mathbf{x}^*) - L\|\mathbf{z} - \mathbf{x}^*\|\big)\frac{2}{L^{\max} + 2\gamma}. \tag{2.79}$$

Since, from (2.77) and (2.78), we have

$$\zeta(\mathbf{x}^*) \le \nabla_i f(\mathbf{z}) + L\|\mathbf{z} - \mathbf{x}^*\|, \tag{2.80}$$

then, using (2.79), we obtain

$$\|\mathbf{z} - \mathbf{x}^*\| \le \frac{2\nabla_i f(\mathbf{z})}{L^{\max} + 2\gamma}.$$

Taking into account that $L_i \le L^{\max}$ and recalling that $x_i^* = l_i$, it follows that (2.75) holds. To prove (2.76), from (2.73b) and the definition of ν_k^i we can write

$$\nu_k^i \le \max_{j=1,\dots,n} \tilde{\alpha}_k^j \le \big(\zeta(\mathbf{x}^*) - L\|\mathbf{y}_k^i - \mathbf{x}^*\|\big)\frac{2}{L^{\max}}.$$

Using (2.80) with $\mathbf{z} = \mathbf{y}_k^i$ and the fact that $L_i \le L^{\max}$, we thus get (2.76).

In view of (2.76) and Proposition 2.4.20, it follows that

$$\mathbf{d}_k^i = -\mathbf{e}_i, \tag{2.81}$$

that is, $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i - \alpha_k^i \mathbf{e}_i$. Using $\mathbf{z} = \mathbf{y}_k^i$ in (2.75), we also have

$$\alpha_{\max} = |(\mathbf{y}_k^i)_i - l_i| \le \frac{2\nabla_i f(\mathbf{y}_k^i)}{L_i + 2\gamma}, \tag{2.82}$$

where $\alpha_{\max}$ is the largest feasible step-size along the direction $\mathbf{d}_k^i$ at $\mathbf{y}_k^i$. So, if $\alpha_{\max} = 0$, then $\alpha_k^i = 0$ and, using (2.81), we have $(\mathbf{y}_k^{i+1})_i = (\mathbf{y}_k^i)_i = l_i$, thus proving (2.74). If $\alpha_{\max} > 0$, using Proposition 2.4.18, (2.81) and (2.82), it follows that a sufficient decrease of f along $\mathbf{d}_k^i$ is obtained with the first step-size $\bar{\alpha}$ used in the line-search, that is, the condition at line 4 of the `Bounded DF-Linesearch` is satisfied. Now, consider the extrapolation phase in the line-search procedure, that is, lines 12–15 of the `Bounded DF-Linesearch`. Recalling (2.81), each step-size $\omega = \min\{\alpha/\delta, \alpha_{\max}\}$ is such that $\alpha \le \omega \le (\mathbf{y}_k^i)_i - l_i$, that is,

$$0 \le \omega - \alpha \le (\mathbf{y}_k^i + \alpha\mathbf{d}_k^i)_i - l_i.$$

So, from (2.81) and the fact that $x_i^* = l_i$, it follows that $\|\mathbf{y}_k^i + \alpha\mathbf{d}_k^i - \mathbf{x}^*\| \le \|\mathbf{y}_k^i - \mathbf{x}^*\|$. So, we can apply (2.75) with $\mathbf{z} = \mathbf{y}_k^i + \alpha\mathbf{d}_k^i$ and then we obtain

$$0 \le \omega - \alpha \le (\mathbf{y}_k^i + \alpha\mathbf{d}_k^i)_i - l_i \le \frac{2\nabla_i f(\mathbf{y}_k^i + \alpha\mathbf{d}_k^i)}{L_i + 2\gamma}$$

for every step-size ω used in the extrapolation. Then, using Proposition 2.4.18 with $\mathbf{x} = \mathbf{y}_k^i + \alpha\mathbf{d}_k^i$, $s = \omega - \alpha$ and $\mathbf{d}_k^i$ as in (2.81), it follows that

$$f(\mathbf{y}_k^i + \omega\mathbf{d}_k^i) \le f(\mathbf{y}_k^i + \alpha\mathbf{d}_k^i) - \gamma(\omega - \alpha)^2.$$

Namely, a sufficient decrease of f is obtained with all step-sizes used in the extrapolation and we quit when we get the largest feasible step-size, meaning that $(\mathbf{y}_k^{i+1})_i$ will be at the lower bound l_i and thus proving (2.74). □

Note that Theorem 2.4.21 establishes finite identification for any limit point of $\{\mathbf{x}_k\}$, thus not requiring the convergence of the whole sequence. Note also, in the proof of Theorem 2.4.21, the crucial role played by the extrapolation in the line-search procedure. Loosely speaking, when we are sufficiently close to a stationary point, expanding the step-size allows us to hit the lower or the upper bound, provided the strict complementarity condition holds. This guarantees to identify all the variables satisfying the strict complementarity after a finite number of iterations.

Now, let us point out a useful property for the limit points of $\{\mathbf{x}_k\}$. To this aim, let us define $\mathcal{X}^\star$ as the set of all limit points of $\{\mathbf{x}_k\}$, i.e.,

$$\mathcal{X}^\star := \{\mathbf{x} \colon \exists \mathcal{K}^{\mathbf{x}} \subseteq \mathbb{N} \text{ such that } \lim_{k\to\infty, k\in\mathcal{K}^{\mathbf{x}}} \mathbf{x}_k = \mathbf{x}\}.$$

The following result roughly states that, if for all $\mathbf{x} \in \mathcal{X}^\star$ there does not exists i such that x_i violates the strict complementarity, then either all x_i lie on the same bound or they all are strictly feasible.

Proposition 2.4.22. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by `LAM` using the `Bounded DF-Linesearch` and consider an index $i \in \{1, \dots, n\}$. Assume that there is no $\mathbf{x} \in \mathcal{X}^\star$ such that $i \in \mathscr{Z}(\mathbf{x}) \setminus \mathscr{Z}^+(\mathbf{x})$. Then,*

- *if there exists $\mathbf{x}^* \in \mathcal{X}^\star$ such that $i \in \mathscr{Z}^+(\mathbf{x}^\star)$, we have that $x_i = x_i^*$ for all $\mathbf{x} \in \mathcal{X}^\star$;*
- *otherwise, $x_i \in (l_i, u_i)$ for all $\mathbf{x} \in \mathcal{X}^\star$.*

Proof. We limit ourselves to show only the first point since the second one can be obtained as a logical consequence. From Theorem 2.4.12 and recalling Ostrowski's theorem [13], the set $\mathcal{X}^\star$ of limit points of the sequence $\{\mathbf{x}_k\}$ is a connected set. Now, let us consider any two points in $\mathcal{X}^\star$, say $\bar{\mathbf{x}} \in \mathcal{X}^\star$ and $\bar{\mathbf{y}} \in \mathcal{X}^\star$, such that $\bar{\mathbf{x}} \ne \bar{\mathbf{y}}$, and $\bar{y}_i \in \{l_i, u_i\}$. Since $\mathcal{X}^\star$ is connected, there exists a continuous function $\rho : [a, b] \to \mathbb{R}^n$ such that $\rho(a) = \bar{\mathbf{x}}$, $\rho(b) = \bar{\mathbf{y}}$ and $\rho(t) \in \mathcal{X}^\star$, i.e., $\rho(t)$ is stationary, for all $t \in [a, b]$. Let us assume, without loss of generality, that $\bar{y}_i = l_i$ (the proof for the case $\bar{y}_i = u_i$ is identical, except for minor changes). By contradiction, now assume that $\bar{x}_i > l_i$. Since $\rho(a)_i = \bar{x}_i > l_i$ and $\rho(b)_i = \bar{y}_i = l_i$, then there exists $\bar{t} \in (a, b]$ such that $\rho(t)_i > l_i$ for all $t \in [a, \bar{t})$

and $\rho(\bar{t})_i = l_i$. Furthermore, by the stationarity conditions given in Definition 2.4.16, we have that $\nabla_i f(\rho(t)) = 0$ for all $t \in [a, \bar{t})$ and $\nabla_i f(\rho(\bar{t})) > 0$, where the last inequality follows from the stated hypothesis. Then, by continuity of $\nabla f(\cdot)$, a scalar $\hat{t} \in (a, \bar{t})$ exists such that $\nabla_i f(\rho(t)) > 0$ for all $t \in (\hat{t}, \bar{t}]$. The latter is a contradiction since $\nabla_i f(\rho(t)) = 0$ for all $t \in (\hat{t}, \bar{t})$. ☐

Applying the above proposition for all indices $i \in \{1, \dots, n\}$, the following result immediately follows, enforcing the finite active-set identification property established in Theorem 2.4.21 when all the limit points of $\{\mathbf{x}_k\}$ are non-degenerate.

Corollary 2.4.23. *Let $\{\mathbf{x}_k\}$ be the sequence of points produced by* `LAM` *using the* `Bounded DF-Linesearch` *and assume that every $\mathbf{x} \in \mathcal{X}^\star$ is non-degenerate. Then, for any pair $\mathbf{x}', \mathbf{x}'' \in \mathcal{X}^\star$, we have $\mathscr{Z}(\mathbf{x}') = \mathscr{Z}(\mathbf{x}'')$ and $x'_i = x''_i$ for all $i \in \mathscr{Z}(\mathbf{x}')$.*

2.5 Conclusions

In this chapter we analysed the worst case complexity of line-search based DFO algorithms for the unconstrained and bound-constrained optimization of a black-box objective function, and we investigated further properties of the bounded case regarding the identification of the active set.

First, we considered `LAM` for unconstrained problems. We have managed to show that the algorithm model takes at most $\mathcal{O}(n\epsilon^{-2})$ iterations and $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations to drive the norm of the gradient below ϵ and that produces at most $\mathcal{O}(n^2\epsilon^{-2})$ iterations where the norm of the gradient is above ϵ. Note that in the paper [20], two slight variations of `LAM` are proposed to generalize the results, where one has better and one has worse dependency on the dimension of the problem for the worst-case complexity bounds with respect to `LAM`. A brief numerical experimentation in the same paper has shown that the algorithm with the best complexity results presents the lowest efficiency, while the algorithm with the worst complexity results presents the best performance.

Furthermore, we have analyzed a derivative-free line-search method for bound-constrained problems. We have first provided complexity results. In more details, given a threshold $\epsilon > 0$, we have shown that the criticality measure $\chi(\mathbf{x}_k)$ (which vanishes at stationary points) falls below ϵ after at most $\mathcal{O}(n\epsilon^{-2})$ iterations, requiring at most $\mathcal{O}(n^2\epsilon^{-2})$ function evaluations. These bounds match those obtained for (deterministic) direct search [51] and model-based [59] methods. Additionally, we have established an upper bound of $\mathcal{O}(n^2\epsilon^{-2})$ on the *total* number of iterations where $\chi(\mathbf{x}_k) \geq \epsilon$. The latter result is obtained thanks to the extrapolation strategy used in the proposed line-search, allowing us to upper bound $\chi(\mathbf{x}_k)$ on both successful and unsuccessful iterations.

In the last part of the paper, we have considered the active-set identification property of the proposed method, i.e., the ability to detect the variables lying on the lower or the upper bounds for the final solutions. We have shown that, in a finite number of iterations, the algorithm identifies the active constraints satisfying the strict complementarity condition. This property is also obtained by exploiting the extrapolation used in the proposed line-search, allowing the step-size to expand until we hit the boundary of the feasible set.

Finally, some topics for future research can be envisaged. In particular, under convexity assumptions, the worst-case complexity of the algorithm might be tightened, in order to match the results given in [39], and a bound on the maximum number of iterations required to identify the active constraints might be given. We wish to report more results in future works.

Chapter 3

A mixed penalty-barrier method for nonlinear constrained optimization*

The work in this chapter originates from an interest in exploring alternative strategies for handling constraints in DFO, particularly within BBO problems involving general nonlinear constraints. Building on previous research that successfully employed a penalty method to handle nonlinear inequalities from the exterior, we wondered if adopting a different strategy—one that operates from within the feasible region—might yield improved constraint handling, particularly for unrelaxable constraints. This led us to consider a logarithmic barrier approach, which restricts the algorithm's search to the interior of the feasible region while enforcing constraints more rigidly. By leveraging this interior approach, we aimed to create a mixed penalty-barrier method (MPB) capable of addressing both inequality and equality constraints, the latter being an aspect less frequently addressed in existing DFO approaches.

To simplify initial analysis, we assumed convexity for the inequality constraints and began working towards convergence to stationary points under this assumption. However, as we pursued this convergence analysis, especially to ensure boundedness in the sequences of multipliers, it became evident that the penalty parameter's updating rule proposed in the previous research required adjustment. Developing this revised updating rule became a critical feature of our method. Meanwhile, the convexity assumption for inequality constraints introduced its own challenges: removing this assumption required overcoming two main obstacles. First, the lack of convexity allows for infeasible points between strictly feasible ones, where the merit function may be undefined, preventing us from directly applying the mean value theorem. Secondly, without convexity, the merit function is not Lipschitz continuous near the boundary of the feasible region. This issue affects convergence since points generated by the algorithm may approach this boundary at varying speeds, impacting the convergence of multipliers. The solution involved two key developments: the use of Proposition 3.2.8, and the proof of the existence of a constant, c_2, in Proposition 3.4.3, relying on an adjusted updating rule that guarantees step-sizes tend to zero faster than the squared values of inequality constraints along a specific subsequence.

Our algorithm, named `LOG-DFL`, was then tested against `NOMAD` in a series of numerical experi-

---

[*] The work presented in this chapter has been developed in two papers. The first, *"An interior point method for nonlinear constrained derivative-free optimization"* [21], co-authored with Prof. Giampaolo Liuzzi ("Sapienza" University of Rome), Prof. Stefano Lucidi ("Sapienza" University of Rome). The second, *"Nonlinear Derivative-free Constrained Optimization with a Mixed Penalty-Logarithmic Barrier Approach and Direct Search"* [19], co-authored with Ph.D. Everton José Da Silva ("NOVA" School of Science and Technology), Prof. Ana Luísa Custódio ("NOVA" School of Science and Technology), Prof. Giampaolo Liuzzi (–).

ments. During these trials, we recognized a limitation in that `LOG-DFL` required the initial solution to strictly satisfy all inequality constraints. To broaden its applicability, we extended the merit function, allowing inequalities to be grouped based on constraint type: some managed by the logarithmic barrier, others by the penalty term also used for equality constraints. This extended approach enabled `LOG-DFL` to run even when starting with partially infeasible solutions and yielded competitive results against `NOMAD`. These observations highlighted areas for potential improvement, such as integrating surrogate models for enhanced performance and leveraging information from infeasible points, similar to techniques used by `NOMAD`. Furthermore, although effective, the extended penalty-barrier function was only applied heuristically in our initial trials, suggesting the need for a more formalized theoretical framework for this function. The work regarding `LOG-DFL` is presented in Section 3.2.

Building on these insights, we initiated a second phase of work aimed at addressing these limitations. We based this phase on the `SID-PSM` algorithm, a GPS method that utilizes polynomial models to enhance search and poll steps. This foundation allowed us to apply the MPB method more robustly, incorporating information from infeasible points and building surrogate models. Unlike the initial work where only bound constraints were directly addressed, this second phase explored handling general linear inequalities directly, taking advantage of pattern search's adaptive direction adjustments in the Poll Step. We adapted our theoretical results from the line-search framework to this DS setting, establishing a rigorous foundation for the extended penalty-barrier function and integrating strategies for handling general linear constraints.

Our final algorithm, `LOG-DS`, demonstrated promising improvements in numerical experiments, outperforming the original `SID-PSM`, which relies on the extreme barrier for constraints. A comparative analysis against `LOG-DFL` and `NOMAD` confirmed `LOG-DS`'s competitive performance, further supporting the potential of our mixed penalty-barrier approach in DFO with complex constraint requirements. This second line of work, detailed in Section 3.3, ultimately advances our understanding of DFO's capabilities, providing insights into handling both interior and exterior constraint enforcement, along with strategies for leveraging infeasible information and surrogate models within a robust theoretical framework.

3.1 Introduction

In this chapter we consider the nonlinear constrained minimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^n} \ & f(\mathbf{x}), \\ \text{s.t.} \ & g(\mathbf{x}) \leq 0, \\ & h(\mathbf{x}) = 0, \end{aligned} \tag{P2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^q$, and $\mathcal{X}$ is a set of linear constraints. We assume that f, g and h are continuously differentiable functions even though their derivatives can be neither calculated nor explicitly approximated. This situation is quite frequent in many real-world problems, for instance those arising in industrial engineering, chemical processes optimization, and many others, where problem function values are computed by means of complex simulation programs. Such programs can be viewed as a black-box that given the values of the decision variables returns the results of the simulation in output. Thus, the analytic expressions of the problem functions are unavailable to the optimizer. As a consequence, derivatives are not available or, at the very least, they are untrustworthy.

Many algorithms have been proposed for the solution of constrained BBO problems. In particular, in [67] the use of an augmented Lagrangian function in connection with a pattern search algorithm has been proposed. In [73] a sequential penalty derivative-free line-search approach has been studied, whereas in [71] the use of a nonsmooth exact penalty function has been proposed. A mesh adaptive direct search method, namely `NOMAD`, has been firstly introduced and analyzed in [4] to solve constrained black-box problems by using an extreme penalty function to manage general and hidden constraints. More recently, in [33], a derivative-free optimization method has been proposed for the solution of structured black-box problems such as e.g. those with constraints of a special structure. The proposed method is able to take advantage of the special structure of the constraints to achieve fast convergence.

When unrelaxable constraints are present in the definition of the problem, they can be typically managed by a so-called *extreme* penalty approach (see e.g. [9]). In particular, an objective function value of $+\infty$ is assigned to points that are infeasible with respect to one or more unrelaxable constraints. However, it should also be mentioned that such penalization strategy, by making the objective function discontinuous on the boundary of the feasible region, introduces many difficulties and ill-conditioning in the problem. As a result, solving the problem could become impractical or the computed solution could be far away from the real one.

A possible way of handling the above mentioned difficulty, consists in the use of some sort of "*interior*" penalization, meaning that the penalization modifies the landscape of the objective function in the interior of the feasible region by adding to the objective function terms that gradually tend to $+\infty$ as the points approach the boundary of the feasible region (see e.g. [46, 34, 80]). This approach might look similar to the one of using an *extreme* barrier, especially when the interior penalty function is defined $+\infty$ outside the feasible region. However the behavior of an interior penalty function approaching the boundary of the feasible set can avoid the difficulties deriving from the discontinuities introduced by an *extreme* penalty approach. For optimization problems where first order information on the objective and constraint functions is available, interior penalty functions have been proved to be an effective tool provided that the sequence of the values of the penalty parameter and the sequence of levels of approximation for the minimization of the penalty function are efficiently connected.

The aim of Section 3.2 is twofold. On one side we extend BM to optimization problems where first order information on the objective and constraint functions is not available. This approach constitutes a viable alternative to the use of the extreme penalty approach when there are unrelaxable inequality constraints. The analysis is inspired by the infeasible penalization technique proposed in [73]. Introducing an alternative feasible method is relevant in the BBO framework where an infeasible approach might not be applicable.

The present work has many similarities with [73], but there are some difficulties that had to be overcome. First of all, to our knowledge, this is the first time a method using a logarithmic barrier is proposed in the field of DFO. The logarithmic barrier alone already introduces many criticalities: the lack of Lipschitz continuity of the gradient of the merit function with respect to the whole feasible set, the discontinuity of the merit function on the boundary of the feasible region, and managing the boundedness of the Lagrangian multipliers, coming from the gradient of the merit function, where the value of the constraints appear at the denominator. Furthermore, in the present work we also consider equality constraints and the interaction between the logarithmic barrier and the exterior penalty. The two different penalty approaches have different purposes, while the exterior penalty tries to push towards feasibility increasing the level of penalization, the logarithmic barrier tries to "release" the penalization as the points are approaching the boundary of the feasible region. Since both levels of penalization are handled by a penalty parameter, finding a way to make both of them work together was a matter of our interest.

In particular, a LS DFO algorithm is proposed. Its main aspect is the definition of a new barrier parameter updating rule which efficiently exploits the information obtained by the sampling of the penalty function performed by the algorithm. In this work we show how such updating rule, even if it might look similar to the one proposed in [73], has wider implications, and it allows to handle both the challenges of the logarithmic barrier and those of the coexistence of the two different penalization strategies. Indeed, we prove that all the accumulation points of a specific sequence of iterates generated by our method satisfying the MFCQ conditions are KKT-stationary. The MFCQ conditions are standard [13] to be assumed when dealing with nonlinearly constrained optimization problems. Furthermore we consider all the functions to be continuously differentiable, and we assume the set $\mathcal{X}$ of bound constraints (linear inequalities are treated within the MPB) to be compact, which is reasonable in practical engineering problems. Finally, we make use of Assumption 3.1, which is necessary when dealing with non-relaxable constraints and when proposing a feasible method. Note that we do not assume convexity over any of the problem functions. Furthermore, a preliminary implementation of the algorithm has shown a very promising numerical behavior on a relatively large set of test problems from the literature when compared to a state-of-the-art solver.

Moreover, we show the algorithm to be working well in practice when compared to the state-of-the-art solver `NOMAD`. When considering problems satisfying the assumptions considered in the theoretical analysis, our method outperforms `NOMAD`. We also provide a numerical evidence for the method to be comparable to `NOMAD` when a wider class of problems is considered, indeed we propose a slight modification of the merit function that allows to handle relaxable inequalities that are violated at the starting point, so that problems not satisfying Assumption 3.1 can be solved. The modification is then theoretically studied in Section 3.3.

The goal of Section 3.3 is to extend the framework proposed in Section 3.2 to DS methods, to provide a theoretical foundation for the modification of the merit function mentioned above, along with explicit treatment not only of the bounds on the variables, but also of general linear inequalities. Our contribution provides a much greater flexibility to handle general constraints, allowing to solve a broader class of challenges where the different nature of constraints can be properly addressed.

Under the same assumptions used in Section 3.2, and without requiring a starting solution which is strictly feasible with respect all the nonlinear inequality constraints, we prove again that all the accumulation points of a specific sequence generated by our method, and satisfying the MFCQ conditions, are KKT-stationary of the problem.

Another motivation for this work arises from the numerical results obtained for `LOG-DFL`, the algorithm proposed in Section 3.2. As stated before, `LOG-DFL` treats all the inequality constraints as unrelaxable, thus, it does not use any information of points violating such constraints. Though, in order to perform a fair comparison, `NOMAD`, even if forced to treat the unrelaxable inequalities with the extreme barrier, was allowed to exploit the values of the objective and the constraints of the same infeasible points. Furthermore, `NOMAD` makes use of quadratic models of the functions to improve the performance, whereas `LOG-DFL` does not. Nevertheless, `LOG-DFL` appears to be very competitive. Therefore, one natural question to be asked is, using such information and exploiting surrogate models, if that would enhance the performance of the MPB approach.

Thus, starting from the `SID-PSM` algorithm [35, 36], an implementation of a GPS method, where polynomial models are used for both the search and the poll steps to improve the numerical performance of the code, `LOG-DS` has been developed, a DS method able to address nonlinear constraints by a MPB approach. In the experiments we carried out and which are showed in the dedicated subsection, `LOG-DS` appears to outperform both `LOG-DFL` and `NOMAD`, especially for problems with the equality constraints.

3.2 LOG-DFL

In this section we propose a DFO method based on line-search techniques which exploits a mixed penalty strategy to solve problem (P2). In particular, in Section 3.2.1 the requirements of the method proposed are shown jointly with some preliminary results. In Section 3.2.2, `LOG-DFL` is introduced and explained in details. In Section 3.2.3 we prove some properties of `LOG-DFL` and the convergence to KKT-stationary points. Finally, In Section 3.2.4 we present some numerical results.

3.2.1 Assumptions and preliminary results

We consider (P2) and we denote by Ω_g the set of feasible points with respect to the inequality constraints, by Ω_h the set defined by the equality constraints, and we assume $\mathcal{X}$ to a be defined only by bound constraints (linear constraints can be considered within Ω_g), that is,

$$\Omega_g = \{\mathbf{x} \in \mathbb{R}^n : g(\mathbf{x}) \leq \mathbf{0}_m\}, \qquad \Omega_h = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) = \mathbf{0}_q\}, \qquad \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\},$$

with $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ and $\mathbf{l} < \mathbf{u}$. By $\mathcal{F}$ we denote the overall feasible set of problem (P2), namely,

$$\mathcal{F} = \Omega_h \cap \Omega_g \cap \mathcal{X}.$$

By definition $\mathcal{X}$ is a compact set so that $\mathcal{F}$ is compact as well. Note that in practical problems one can usually restrict the variables to reasonable bounds, thus the compactness assumption on $\mathcal{X}$ is a viable one.
In [64, 63] a complete taxonomy of constraints for simulation-based optimization has been proposed. In particular, constraints can be classified as either *known* (K) or *hidden* (H), *a priori* (A) or *simulation* (S), *relaxable* (R) or *unrelaxable* (U), *quantifiable* (Q) or *nonquantifiable* (N). In this paper, we assume that the inequality constraints defining set Ω_g are of type: known, simulation, unrelaxable and quantifiable. More specifically, as stated in [64, Definition 1], the degree of feasibility of unrelaxable constraints is always accessible. The degree of violation might or might not be available, but it is always possible to detect when a point is infeasible. Note that whenever one or more unrelaxable constraints are violated, the violation, whether with or without the relative amount, is the only meaningful output, i.e. values of the objective function and feasible constraints are not available. We point out that the method we propose in this section does not require to compute the degree of violation of the constraints defining Ω_g, so that a wider class of practical problems can be addressed.

Throughout this section, we require the following assumption to hold true.

Assumption 3.1.

- *The inequality constraints g_ℓ, $\ell = 1, \dots, m$, are "unrelaxable", that is the objective function f cannot be computed at $\mathbf{x}$ when $g_\ell(\mathbf{x}) > 0$ for (at least one) $\ell \in \{1, \dots, m\}$.*
- *A point $\mathbf{x}_0 \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$ is known.*

In the framework of design optimization, where variables represent geometrical properties of the object to be modeled, an initial feasible project is usually available. The objective to be optimized is usually computed by a simulation program that would fail if geometrical constraints are not satisfied, i.e., the constraints are non-relaxable. For instance, in [23] the optimal design of the hull of the S-175 containership [43] has been considered. Roughly speaking, the problem is that of finding the ship hull that conveys the highest reduction of the heave motion of the ship when advancing in head seas. The ship hull must satisfy a few inequality constraints related in particular to the displacement and the length of the ship. The initial design satisfies these constraints but has a high heave motion. Moreover, if the displacement and length constraints happen to be violated then the simulator cannot compute the heave motion of the ship.

To solve problem (P2), we propose a mixed penalty-barrier method. We make use of a merit function wherein inequality (unrelaxable) constraints are handled by logarithmic barrier terms and equality constraints are addressed by standard exterior penalty terms (see e.g. [45]), i.e.

$$z(\mathbf{x};\rho) = f(\mathbf{x}) - \rho\sum_{\ell=1}^{m} \log\left[-g_\ell(\mathbf{x})\right] + \frac{1}{\rho}\sum_{j=1}^{q}|h_j(\mathbf{x})|^\nu,$$

where $\nu \in (1,2]$. Bound constraints on the variables will be addressed explicitly by the optimization algorithm. Note that in $z(\cdot)$ a single penalty parameter ρ is used to scale the terms relative to both the interior and exterior penalization. It allows us to keep the presentation of the algorithm as simple as possible and to avoid excessive technicalities in the theoretical analysis.

Then, in place of the original problem (P2), we consider the "penalized" problem

$$\begin{aligned} &\min\ z(\mathbf{x};\rho) \\ &s.t.\ \mathbf{x} \in \overset{\circ}{\Omega}_g \cap \mathcal{X} \end{aligned} \tag{C1}$$

Note that for every fixed value of the penalty parameter ρ, $z(x;\rho)$ is continuously differentiable in $\overset{\circ}{\Omega}_g$ under the stated assumptions. In order to obtain a clearer description of the proposed algorithm, we assume that, $z(\mathbf{x};\rho) = +\infty$, for all $\mathbf{x} \in \mathbb{R}^n$ such that $g(\mathbf{x}) \not< 0$. Hence, in place of problem (C1), the following problem can be considered

$$\begin{aligned} &\min\ z(\mathbf{x};\rho) \\ &s.t.\ \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{P3}$$

Within a directional DFO approach it can be particularly useful to exploit the particular structure of $\mathcal{X}$ by dealing directly with the box constraints. The idea is to efficiently sample a merit function that only penalizes nonlinear constraints at points that satisfy the box constraints. To this end, we provide the specific definition of the tangent cone of a point $\mathbf{x}$ with respect to $\mathcal{X}$ and recall some of its properties.

Definition 3.2.1 (cone of feasible directions). *Given a point* $\mathbf{x} \in \mathcal{X}$*, let*

$$T_{\mathcal{X}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n : d_i \geq 0 \ \textit{if}\ x_i = l_i,\ d_i \leq 0\ \textit{if}\ x_i = u_i,\ i = 1,\dots,n\}$$

be the cone of feasible directions at x *with respect to the simple bound constraints.*

Then, let us define the set of unit vectors

$$\mathcal{E} = \{\pm\mathbf{e}^1,\dots,\pm\mathbf{e}^n\},$$

where $\mathbf{e}^i$, $i = 1,\dots,n$, is the i-th unit coordinate vector.

Now we recall two results from [70] and [73] concerning the relation between set $\mathcal{E}$ and $T_{\mathcal{X}}(\mathbf{x})$. In particular, point (a) of the following proposition shows that $\mathcal{E}$ contains the generators of the cone of feasible directions $T_{\mathcal{X}}(\mathbf{x})$ at any point $\mathbf{x} \in \mathcal{X}$; point (b) proves that the feasible directions (with respect to the bounds) at any point are also feasible for points sufficiently close to it.

Proposition 3.2.2.

(a) Let $\mathbf{x} \in \mathcal{X}$*. It holds that*

$$cone\{\mathcal{E} \cap T_{\mathcal{X}}(\mathbf{x})\} = T_{\mathcal{X}}(\mathbf{x}). \tag{3.1}$$

(b) Let $\{\mathbf{x}_k\}$ *be a sequence of points such that* $\mathbf{x}_k \in \mathcal{X}$ *for all* k*. Assume further that* $\mathbf{x}_k \to \bar{\mathbf{x}}$ *for* $k \to \infty$*. Then, given any direction* $\bar{\mathbf{d}} \in T_{\mathcal{X}}(\bar{\mathbf{x}})$*, there exists a scalar* $\bar{\beta} > 0$ *such that, for sufficiently large* k*, we have*

$$\mathbf{x}_k + \beta\bar{\mathbf{d}} \in \mathcal{X} \quad \forall\beta \in [0, \bar{\beta}].$$

The presence of the box constraints can also be exploited to introduce particular optimality conditions which highlight the possibility of treating the box constraints in a different way with respect to other constraints.

As usual in the nonlinear context, any necessary optimality conditions need the constraints to satisfy some qualification condition. In particular, we recall the following extended Mangasarian-Fromovitz constraints qualification (EMFCQ) which points out the role of the cone $T_{\mathcal{X}}(\mathbf{x})$.

Definition 3.2.3 (Extended Mangasarian–Fromovitz constraints qualification [13, Proposition 4.3.18])**.** *A point* $\mathbf{x} \in \mathcal{X}$ *is said to satisfy the EMFCQ if the following two conditions are met:*

(a) *there does not exist a nonzero vector* $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_q)$ *such that:*

$$\left(\sum_{i=1}^{q} \alpha_i \nabla h_i(\mathbf{x})\right)^T \mathbf{d} \geq 0, \qquad \forall \mathbf{d} \in T_{\mathcal{X}}(\mathbf{x}); \tag{3.2}$$

(b) *there exists a feasible direction* $\mathbf{d} \in T_{\mathcal{X}}(\mathbf{x})$ *such that:*

$$\nabla g_\ell(\mathbf{x})^T \mathbf{d} < 0 \quad \forall \ell \in \mathcal{I}^+(\mathbf{x}), \quad \nabla h_j(\mathbf{x})^T \mathbf{d} = 0 \quad \forall j = 1, ..., q \tag{3.3}$$

where $\mathcal{I}^+(\mathbf{x}) = \{i : g_i(\mathbf{x}) \geq 0\}$*.*

Note that the above definition is an extension of the well-known Mangasarian-Fromovitz constraint qualification (MFCQ) [77]. Indeed, whenever $T_{\mathcal{X}}(\mathbf{x}) = \mathbb{R}^n$, which happens in particular when either $\mathbf{x} \in \overset{\circ}{\mathcal{X}}$ or $\mathcal{X} = \mathbb{R}^n$, Definition 3.2.3 extends the MFCQ considering also infeasible points.

Now, the Lagrangian function associated with the nonlinear constraints of problem (P2) is

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T g(\mathbf{x}) + \boldsymbol{\mu}^T h(\mathbf{x}).$$

Then, the following proposition report a well-known result (see, for instance, [13]) which states necessary optimality conditions for problem (P2).

Proposition 3.2.4. *Let* $\mathbf{x}^\star \in \mathcal{F}$ *be a local minimum of problem* (P2) *that satisfies the EMFCQ. Then,* $\boldsymbol{\lambda}^\star \in \mathbb{R}^m$*,* $\boldsymbol{\mu}^\star \in \mathbb{R}^q$ *exist such that*

$$\nabla_x L(\mathbf{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)^T (\mathbf{x} - \mathbf{x}^\star) \geq 0 \qquad \forall \mathbf{x} \in \mathcal{X}, \tag{3.4}$$

$$(\boldsymbol{\lambda}^\star)^T g(\mathbf{x}^\star) = 0, \qquad \boldsymbol{\lambda}^\star \geq 0. \tag{3.5}$$

On the basis of Proposition 3.2.4, it is possible to introduce the definition of stationary point for problem (P2).

Definition 3.2.5 (stationary point). *A point* $\mathbf{x}^\star \in \mathcal{F}$ *is said to be a stationary point for problem* $(P2)$ *if vectors* $\boldsymbol{\lambda}^\star \in \mathbb{R}^m$ *and* $\boldsymbol{\mu}^\star \in \mathbb{R}^q$ *exist such that* (3.4) *and* (3.5) *are satisfied.*

3.2.2 LOG-DFL Algorithm

In this section, we describe and analyze the proposed DFO algorithm used to solve the constrained problem (P2).

The algorithm that we propose approximates stationary points of problem (P2) by approximately solving problem (P3) for decreasing values of the penalty-barrier parameter ρ.

The scheme of the algorithm, as it can be easily seen, follows the lines of other line-search based DFO algorithms, see e.g. [74, 73]. According to this approach, sufficient information on the behavior of the merit function $z(\cdot)$ along the feasible directions is extracted by performing suitable samplings along particular sets of directions. Proposition 3.2.2 (see also [60, 74]) shows that, in the case of bound constrained optimization problems, the set $\mathcal{E}$ of the unit coordinate vectors $\mathbf{e}^i$, $i = 1, \dots, n$ is a suitable choice for the set of search directions.

```
                         Algorithm LOG-DFL.
Data. $\mathbf{x}_0 \in \mathcal{X}$ such that $g(\mathbf{x}_0) < 0$, $\rho_0 > 0$, $\gamma > 0$, $\{\delta, \theta\} \subset (0,1)$, $\tilde{\alpha}_0^i > 0$,
      $\nu \in (1,2]$, $\beta > \frac{1}{\nu-1}$ and set $\mathbf{d}_0^i = \mathbf{e}^i$ for $i = 1, \dots, n$.
For $k = 0, 1, 2, \dots$ do (Main iteration loop)
    Step 1. Set $\mathbf{y}_k^1 = \mathbf{x}_k$
        For $i = 1, \dots, n$ do (Exploration of the search directions)
            Forall $\mathbf{p} \in \{\mathbf{d}_k^i, -\mathbf{d}_k^i\}$ do
              Let $b$ be the largest step such that $\mathbf{y}_k^i + b\mathbf{p} \in \mathcal{X}$.
              Set $\hat{\alpha}_k^i = \min\{\tilde{\alpha}_k^i, b\}$
              If $\hat{\alpha}_k^i > 0$, and $z(\mathbf{y}_k^i + \hat{\alpha}_k^i \mathbf{p}; \rho_k) \le z(\mathbf{y}_k^i; \rho_k) - \gamma(\hat{\alpha}_k^i)^2$,
                compute $\alpha_k^i$ by the Expansion Step($\hat{\alpha}_k^i, b, \mathbf{y}_k^i, \mathbf{p}, \gamma, \rho_k; \alpha_k^i$);
                Set $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i + \alpha_k^i \mathbf{p}$, $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $\mathbf{d}_{k+1}^i = \mathbf{p}$ and break
              Else
                Set $\alpha_k^i = 0$, $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i + \alpha_k^i \mathbf{p}$, $\mathbf{d}_{k+1}^i = \mathbf{p}$, $\tilde{\alpha}_{k+1}^i = \delta \tilde{\alpha}_k^i$
            End forall
        Endfor
    Step 2. Set $(g_{min})_k = \min_{\substack{i = 1, \dots, n+1, \\ \ell = 1, \dots, m}} \{|g_\ell(\mathbf{y}_k^i)|\}$
            If $\max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} \le \min\{\rho_k^\beta, (g_{\min})_k^2\}$
            Then choose $\rho_{k+1} = \theta\rho_k$ Else set $\rho_{k+1} = \rho_k$.
    Step 3. Find $\mathbf{x}_{k+1} \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$ such that $z(\mathbf{x}_{k+1}; \rho_{k+1}) \le z(\mathbf{y}_k^{n+1}; \rho_{k+1})$.
Endfor
```

Some comments regarding algorithm `LOG-DFL` are in order. At every iteration k of the method we have the following three main steps:

- (Step 1) exploration of the search directions;
- (Step 2) updating of the penalty-barrier parameter ρ_k;
- (Step 3) selection of the next iterate $\mathbf{x}_{k+1}$.

In **Step 1**, the set of search directions $\mathcal{D}_k = \{\mathbf{d}_k^1, \dots, \mathbf{d}_k^n\}$ is employed. The directions are initialized to the unit coordinate vectors, i.e. $\mathbf{d}_0^i = \mathbf{e}^i$, for $i = 1, \dots, n$, and iteratively updated by the method itself.

Starting from the current iterate $\mathbf{y}_k^1 = \mathbf{x}_k$, all the directions $\mathbf{d}_k^i$, $i = 1, \dots, n$, are sequentially explored. Indeed, the i-th iteration of the inner `for` loop, produces the actual step-size $\alpha_k^i \geq 0$ and the new point for the subsequent $(i+1)$-th iteration, namely $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i \pm \alpha_k^i \mathbf{d}_k^i$. Note that, $\mathbf{y}_k^{i+1} = \mathbf{y}_k^i$ when $\alpha_k^i = 0$. More precisely, the following two cases can happen:

i) $\alpha_k^i > 0$ when "sufficient" decrease is achieved along $\mathbf{d}_k^i$ or $-\mathbf{d}_k^i$;

ii) $\alpha_k^i = 0$ otherwise.

In particular, the step-sizes α_k^i are computed by visiting the i-th coordinate direction and seeking for a sufficiently large movement along either $\mathbf{d}_k^i$ or $-\mathbf{d}_k^i$. To this aim, through the definition of $\hat{\alpha}_k^i$ we decide whether direction $\mathbf{d}_k^i$ is feasible at $\mathbf{y}_k^i$ and if sufficient reduction can be achieved. In this case, $\alpha_k^i \geq \hat{\alpha}_k^i$ is computed by means of an expansion step. If $\mathbf{d}_k^i$ is not feasible or when sufficient reduction cannot be achieved, the opposite direction $-\mathbf{d}_k^i$ is tried (by using the same strategy) before declaring failure and setting $\alpha_k^i = 0$.

Exploring the directions $\mathbf{d}_k^i$ for $i \in \{1, \dots, n\}$, the following rule is adopted to define $\mathbf{d}_{k+1}^i$:

$$\mathbf{d}_{k+1}^i = \begin{cases} \mathbf{d}_k^i & \text{when "sufficient" decrease is achieved along } \mathbf{d}_k^i, \\ -\mathbf{d}_k^i & \text{when "sufficient" decrease is achieved along } -\mathbf{d}_k^i, \\ \mathbf{d}_k^i & \text{otherwise.} \end{cases}$$

Let us further discuss the computation of the actual steps α_k^i. In particular, when $\alpha_k^i > 0$, the step is computed by means of the following `Expansion Step` procedure.

```
Expansion Step ($\hat{\alpha}, b, \mathbf{y}, \mathbf{p}, \gamma, \rho; \alpha$).

Data. $\delta \in (0,1)$ and $b$ the largest step such that $\mathbf{y} + b\mathbf{p} \in \mathcal{X}$.
Step 1. Set $\alpha \leftarrow \hat{\alpha}$.
Step 2. set $\breve{\alpha} \leftarrow \min\{b, \alpha/\delta\}$
Step 3. If $\mathbf{y} + \breve{\alpha}\mathbf{p} \notin \overset{\circ}{\Omega}_g$ return
            Elseif $\breve{\alpha} < b$ and $z(\mathbf{y} + \breve{\alpha}\mathbf{p}; \rho) \leq z(\mathbf{y}; \rho) - \gamma\breve{\alpha}^2$ then
                set $\alpha \leftarrow \breve{\alpha}$
            Elseif $\breve{\alpha} = b$ and $z(\mathbf{y} + \breve{\alpha}\mathbf{p}; \rho) \leq z(\mathbf{y}; \rho) - \gamma\breve{\alpha}^2$ then
                set $\alpha \leftarrow \breve{\alpha}$ and return
            Else (i.e. $z(\mathbf{y} + \breve{\alpha}\mathbf{p}; \rho) > z(\mathbf{y}; \rho) - \gamma\breve{\alpha}^2$)
                return
Step 4. Go to Step 2.
```

The main aim of the `Expansion Step` procedure is to compute a step $\alpha \geq \hat{\alpha}$ such that

1. the new point is strictly feasible, i.e. $\mathbf{y} + \alpha\mathbf{p} \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$;
2. sufficient decrease is obtained w.r.t. $\mathbf{y}$, i.e.

$$z(\mathbf{y} + \alpha\mathbf{p}; \rho) \leq z(\mathbf{y}; \rho) - \gamma\alpha^2.$$

Note that the `Expansion Step` is invoked with a step-size $\alpha = \hat{\alpha}$ and a direction $\mathbf{p}$ such that

$$\mathbf{y} + \alpha\mathbf{p} \in \mathcal{X} \cap \overset{\circ}{\Omega}_g \quad \text{and} \quad z(\mathbf{y} + \alpha\mathbf{p}; \rho) \leq z(\mathbf{y}; \rho) - \gamma\alpha^2. \tag{3.6}$$

Then the *expansion* cycles between Step 2 and Step 4 to try and increase the step-size along $\mathbf{p}$ as much as possible, guaranteeing a sufficient decrease on the objective function. To better understand the expansion cycle, we have to delve into the conditions of Step 3. First, note that the conditions involve the tentative step $\check{\alpha}$. Then the following cases can occur:

1. the tentative step $\check{\alpha}$ is too large up to the point that $\mathbf{y} + \check{\alpha}\mathbf{p} \notin \overset{\circ}{\Omega}_g$ (i.e. the first if condition). Then the expansion cycle terminates producing the last step α which surely satisfies (3.6);
2. the tentative step-size is such that the new point is on the boundary of $\mathcal{X}$, within $\overset{\circ}{\Omega}_g$ and such that sufficient reduction is guaranteed (i.e. the third if condition). In this case, the expansion cycle terminates producing $\alpha = \check{\alpha}$ which satisfies (3.6);
3. the tentative step-size does not achieve sufficient decrease, i.e. $z(\mathbf{y} + \check{\alpha}\mathbf{p}; \rho) > z(\mathbf{y}; \rho) - \gamma\check{\alpha}^2$. In this case, the expansion cycle terminates producing the last step-size α which satisfies (3.6);
4. the tentative step-size is such that sufficient decrease is obtained and there is still room for possible step enlargements (i.e. the second if condition). In this case, the expansion cycle does not terminate but updates the current step α.

Considering the above description, the `Expansion Step` procedure computes also a further step-size, namely $\check{\alpha}$, such that $\alpha < \check{\alpha} \leq \alpha/\delta$. The ideal situation (and the one that will be used in the proof of Proposition 3.2.8) for $\check{\alpha}$ would be

$$\mathbf{y} + \check{\alpha}\mathbf{p} \in \overset{\circ}{\Omega}_g, \quad \text{and} \quad z(\mathbf{y} + \check{\alpha}\mathbf{p}; \rho) > z(\mathbf{y}; \rho) - \gamma\check{\alpha}^2.$$

However, the following cases might occur as well:

i) either $\mathbf{y} + \check{\alpha}\mathbf{p} \notin \overset{\circ}{\Omega}_g$;

ii) or $\check{\alpha} = b$, $\mathbf{y} + \check{\alpha}\mathbf{p} \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$ and

$$z(\mathbf{y} + \check{\alpha}\mathbf{p}; \rho) \leq z(\mathbf{y}; \rho) - \gamma\check{\alpha}^2;$$

In case (i), considering that $\mathbf{y} + \alpha\mathbf{p} \in \overset{\circ}{\Omega}_g$ and exploiting the structure of the barrier function, we will show that $\xi > 0$ exists such that $\alpha < \xi < \check{\alpha}$ and $\mathbf{y} + \xi\mathbf{p} \in \overset{\circ}{\Omega}_g$ and

$$z(\mathbf{y} + \xi\mathbf{p}; \rho) > z(\mathbf{y}; \rho) - \gamma\xi^2.$$

Whereas for case (ii), by exploiting the particular structure of the search directions, we will show that eventually it cannot happen (see the proof of Proposition 3.2.8).

In **Step 2**, the penalty-barrier parameter ρ_k is (possibly) updated. We can say that the main novelty of algorithm `LOG-DFL` resides in this step. In fact, as we shall see in the convergence analysis, the updating rule is crucial to prove convergence of the sequence of iterates toward stationary points of the constrained problem. As we can see, the barrier-penalty parameter updating rule is defined on the basis of two quantities, namely

i) the "maximum step-size" $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$, i.e. the maximum step-size used by the algorithm in the entire inner `for` loop. We recall that this quantity can be roughly considered as a measure of stationarity for the penalty function, see e.g. [74, 60];

ii) the "minimum value" for the non-relaxable inequality constraints $(g_{\min})_k$, i.e. the smallest absolute value of the inequality constraints found in the inner `for` loop, namely: $\min\limits_{\substack{i=1,\dots,n+1,\\ \ell=1,\dots,m}} \{|g_\ell(\mathbf{y}_k^i)|\}$.

These two quantities play a crucial role in the penalty-barrier parameter updating rule and hence in the convergence analysis. More precisely, the algorithm updates the penalty-barrier parameter when the measure of stationarity $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$ is smaller than the smallest value between ρ_k^β and $(g_{\min})_k^2$. In more details, ρ_k is reduced when both the following conditions are satisfied.

i) $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$ is smaller than ρ_k^β;

ii) $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$ is smaller than $(g_{\min})_k^2$.

Condition (i) requires that the measure of stationarity is better than the quality of the approximation performed by the merit function w.r.t. the constrained problem. Indeed, the penalty function approaches the extreme penalty as the penalty parameter ρ approaches zero. On the other hand, condition (ii) requires the step-size used by the algorithm to be sufficiently small in order to drive the iterates towards the boundary of the feasible region. It is worth noting that both conditions imply that the maximum step-size $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$ must go to zero faster than the penalty-barrier parameter (which is required to go to zero in order for the iterates to approach a KKT point in the limit) and than the minimum value for the non-relaxable inequality constraints (in the case the limit point lies on the boundary of the feasible region) respectively.

Condition (i) of the updating rule is the same used in [73] where an exterior penalty approach has been proposed to solve problems with only inequality constraints. The updating rule we propose in our algorithm is more restrictive than the one in [73] because in algorithm `LOG-DFL` we have to take into account the effects of two types of constraint penalization techniques. More specifically, the precision required when solving a subproblem with a given penalty parameter ρ must be sufficiently high, i.e. the maximum step length $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\}$ must be sufficiently small, relative to the current approximation of the constrained problem performed by the merit function and to the proximity of the points generated by the algorithm to the boundary of Ω_g.

Finally, in **Step 3**, the new point $\mathbf{x}_{k+1}$ is selected to be any point which is better than the one produced by the inner for loop in Step 1, i.e. $\mathbf{y}_k^{n+1}$. In particular, since we require that $z(\mathbf{x}_{k+1}; \rho_{k+1}) \leq z(\mathbf{y}_k^{n+1}; \rho_{k+1})$, a viable choice could be $\mathbf{x}_{k+1} = \mathbf{y}_k^{n+1}$. **Step 3** is especially useful considering the iterations where the penalty parameter ρ_k is updated. Those are iterations where the shape of the merit function gets modified, and the current iterate might not be the best one among the ones previously generated by the algorithm. Thus, when that happens, it is possible to select the point which provides the best value of the current merit function as the current iterate. Furthermore, the freedom in the selection of the new point $\mathbf{x}_{k+1}$ can be exploited in many other ways. For example, at the end of the iteration, we could try to build quadratic models of the objective and constraint functions to help define a tentative point $\tilde{\mathbf{x}}_{x+1}$. Such a point could then be selected as the new point if the merit function value does not deteriorate with respect to $\mathbf{y}_k^{n+1}$.

3.2.3 Convergence analysis

This section is devoted to the analysis of the convergence properties of the proposed algorithm.

First of all we have to show that algorithm `LOG-DFL` is well-defined, that is it produces the infinite sequences of points and scalars necessary to prove asymptotic convergence. In particular, since the `Expansion Step` procedure is possibly invoked at iteration k when exploring directions $\pm\mathbf{d}_k^i$, $i = 1, \dots, n$, we must guarantee that the procedure never infinitely cycles. Indeed, in the following proposition, we prove that the `Expansion Step` procedure always returns a step α.

Proposition 3.2.6. *The* `Expansion Step` *is well defined, i.e. it always returns a step* α.

Proof. We proceed by contradiction and assume that the procedure infinitely cycles. If that is the case, it produces an infinite sequence of values $\{\breve{\alpha}_j\}$. By the instructions:

$$\breve{\alpha}_j = \frac{\alpha}{\delta^j},$$

which contradicts $\breve{\alpha} < b$. □

The next theorem ensures that the updating rule of the algorithm produces sequences $\{\rho_k\}$, $\{\tilde{\alpha}_k^i\}$, and $\{\alpha_k^i\}$ of penalty parameter values, tentative and actual step-sizes, respectively, which tend to zero. This result is of paramount importance for two reasons:

1. the parameter ρ_k multiplies the log-barrier terms of the merit function and must indeed go to zero to allow us to prove convergence to stationary points of the original problem;

2. the step-sizes going to zero means that the algorithm performs finer and finer sampling of the merit function along the search directions, thus recovering the unavailable first order information in the limit.

Theorem 3.2.7. *Let* $\{\rho_k\}$ *be the sequence of barrier-penalty parameters and* $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$, $i = 1, 2, \dots, n$, *be the sequences of tentative and actual step-sizes produced by Algorithm* `LOG-DFL`. *Let* $\mathcal{K}_\rho = \{k \in \mathbb{N} : \rho_{k+1} < \rho_k\}$. *Then:*

1. $\lim_{k\to\infty} \rho_k = 0$;

2. $\lim_{k\to\infty, k\in\mathcal{K}_\rho} \max_{i=1,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} = 0$.

Proof. We first prove point (1). By the instructions of the algorithm, $\{\rho_k\}$ is a monotonically non-increasing sequence of positive numbers. Hence, it is convergent to a limit $\bar{\rho} \geq 0$. Then, we proceed by contradiction and assume that $\bar{\rho} > 0$. That means, for k sufficiently large, ρ_k is no longer updated. Hence, we can assume that ρ_k stays fixed, i.e. $\rho_k = \bar{\rho}$, definitely, i.e. the test at step 2 of Algorithm `LOG-DFL` is no longer satisfied that is

$$\max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} > \min\{\bar{\rho}^\beta, (g_{\min})_k^2\}. \tag{3.7}$$

By the instructions of Algorithm `LOG-DFL`, we have that, for all k sufficiently large,

$$z(\mathbf{x}_{k+1};\bar{\rho}) \leq z(\mathbf{y}_k^{n+1};\bar{\rho}) \leq \cdots \leq z(\mathbf{y}_k^1;\bar{\rho}) = z(\mathbf{x}_k;\bar{\rho}). \tag{3.8}$$

Hence,

$$\lim_{k\to\infty} z(\mathbf{x}_k;\bar{\rho}) = \bar{z} < +\infty. \tag{3.9}$$

Then, since $\rho_k \to \bar{\rho} > 0$, we can use the same arguments as in [73, Proposition 5, point (ii)]. In particular, for each index $i = 1,\dots,n$, let us split the iteration sequence into two sets $\mathcal{K}_1 = \{k : \alpha_k^i = 0\}$ and $\mathcal{K}_2 = \{k : \ \alpha_k^i > 0\}$. By (3.8), we have that

$$z(\mathbf{x}_{k+1};\bar{\rho}) \leq z(\mathbf{y}_k^i + \alpha_k^i \mathbf{d}_k^i) \leq z(\mathbf{y}_k^i;\bar{\rho}) - \gamma(\alpha_k^i)^2 \leq z(\mathbf{x}_k;\bar{\rho}).$$

If $\mathcal{K}_2$ is infinite, taking the limit for $k \to \infty, k \in \mathcal{K}_2$ in the above relation and taking into account (3.9), we obtain

$$\lim_{k\to\infty, k\in\mathcal{K}_2} \alpha_k^i = 0.$$

Hence, since $\alpha_k^i = 0$, for all $k \in \mathcal{K}_1$, we have that

$$\lim_{k\to\infty} \alpha_k^i = 0. \tag{3.10}$$

Now, recall that for all $k \in \mathcal{K}_1$, we have $\tilde{\alpha}_{k+1}^i = \delta\tilde{\alpha}_k^i$, whereas all $k \in \mathcal{K}_2$ are such that $\tilde{\alpha}_{k+1}^i = \alpha_k^i$. Hence, if $\mathcal{K}_2$ is infinite, from (3.10), we have

$$\lim_{k\to\infty, k\in\mathcal{K}_2} \tilde{\alpha}_{k+1}^i = 0. \tag{3.11}$$

On the other hand, if $\mathcal{K}_1$ is infinite, for each $k \in \mathcal{K}_1$, let us denote by $m(k)$ the biggest index such that $m(k) < k$ and $m(k) \in \mathcal{K}_2$. Then, by the instructions of the algorithm, we can write

$$\tilde{\alpha}_{k+1}^i = \delta^{k+1-m(k)}\tilde{\alpha}_{m(k)}^i \leq \tilde{\alpha}_{m(k)}^i.$$

Then, from (3.11), we obtain

$$\lim_{k\to\infty, k\in\mathcal{K}_1} \tilde{\alpha}_{k+1}^i = 0. \tag{3.12}$$

Then, from (3.11) and (3.12), we finally obtain

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0.$$

Repeating the above reasoning for all indices $i = 1,\dots,n$, we get

$$\lim_{k\to\infty} \tilde{\alpha}_k^i = 0 \qquad \textit{for } i = 1,\dots,n, \tag{3.13}$$

$$\lim_{k\to\infty} \alpha_k^i = 0 \qquad \textit{for } i = 1,\dots,n, \tag{3.14}$$

Now, recalling (3.7) and the fact that $\rho_k = \bar{\rho}$ for all k sufficiently large, we have that:

$$\lim_{k\to\infty} (g_{min})_k = 0.$$

Given the definition of $(g_{min})_k$ in the algorithm and the fact that the number of constraints m and of the variables n are both finite, an infinite index set $\mathcal{K}'' \subseteq \{0,1,\dots\}$ exists such that

$$(g_{min})_k = |g_{\bar{\ell}}(\mathbf{y}_k^{\bar{i}})|,$$

for some $\bar{\jmath} \in \{1, \dots, m\}$ and $\bar{\imath} \in \{1, \dots, n+1\}$. By the instructions of the algorithm, we also have

$$\mathbf{y}_k^{\bar{\imath}} = \mathbf{x}_k + \sum_{\ell=1}^{\bar{\imath}-1} \alpha_k^\ell \mathbf{d}_k^\ell.$$

Now, since $\mathbf{x}_k \in \mathcal{X}$ then, a subset of indices $\mathcal{K}''' \subseteq \mathcal{K}''$ exists such that

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}'''}} \mathbf{x}_k = \bar{\mathbf{x}}$$
$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}'''}} \mathbf{y}_k^{\bar{\imath}} = \bar{\mathbf{x}},$$

and

$$\lim_{\substack{k\to\infty,\\ k\in K'''}} |g_{\bar{\ell}}(\mathbf{x}_k)| = 0$$

i.e. $\bar{\mathbf{x}} \in \partial\Omega_g \cap \mathcal{X}$. Hence, we have that

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}'''}} z(\mathbf{x}_k, \bar{\rho}) = +\infty$$

This is a contradiction with (3.9) and concludes the proof of point (1).
Concerning point (2), let us assume for the sake of contradiction that a subset of iterations $\mathcal{K} \subseteq \mathcal{K}_\rho$ exists such that, for all $k \in \mathcal{K}$,

$$\max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} \geq \bar{\alpha}.$$

Since $\rho_k \to 0$, it also results that $\lim_{k\to\infty, k\in\mathcal{K}} \rho_k = 0$, i.e. a further set $\bar{\mathcal{K}} \subseteq \mathcal{K}$ exists such that $\lim_{k\to\infty, k\in\bar{\mathcal{K}}} \rho_k = 0$ and, since $\bar{\mathcal{K}} \subseteq \mathcal{K} \subseteq \mathcal{K}_\rho$, for all $k \in \bar{\mathcal{K}}$,

$$\bar{\alpha} \leq \max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} \leq \min\{\rho_k^\beta, (g_{min})_k^2\} \leq \rho_k^\beta.$$

This last relation contradicts point (i) and concludes the proof. □

Now, we introduce the following set of iteration indices

$$\mathcal{K}_\rho = \{k : \ \rho_{k+1} < \rho_k\}. \tag{3.15}$$

Note that, by virtue of Theorem 3.2.7, $\mathcal{K}_\rho$ is an infinite index set.

In the next proposition we report a technical result needed to show the convergence properties of the algorithm. It points out that, eventually, the algorithm performs suitable samplings of the merit function along all the generators of the cone of feasible directions at any limit point $\bar{\mathbf{x}}$.

Proposition 3.2.8. *Let* $\{\mathbf{x}_k\}$, $\{\rho_k\}$, *and* $\{\mathbf{y}_k^i\}$, $i = 1, \dots, n+1$, *be the sequences produced by Algorithm* `LOG-DFL` *and let* $\{\mathbf{x}_k\}_{\mathcal{K}_\rho^{\mathbf{x}}}$, $\mathcal{K}_\rho^{\mathbf{x}} \subseteq \mathcal{K}_\rho$, *be a subsequence converging to the point* $\bar{\mathbf{x}}$. *Then, for all* $\mathbf{d}^i \in \mathcal{E} \cap T_{\mathcal{X}}(\bar{\mathbf{x}})$, *there exist vectors* $\mathbf{z}_k^i$ *and scalars* $\xi_k^i > 0$ *such that:*
for $k \in \mathcal{K}_\rho^{\mathbf{x}}$ *sufficiently large,*

$$\mathbf{z}_k^i + t\xi_k^i \mathbf{d}^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g, \quad \forall\ t \in [0,1], \tag{3.16}$$
$$z(\mathbf{z}_k^i + \xi_k^i \mathbf{d}^i; \rho_k) > z(\mathbf{z}_k^i; \rho_k) - o(\xi_k^i); \tag{3.17}$$

and,

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \xi_k^i = 0, \tag{3.18}$$

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \|\mathbf{z}_k^i - \mathbf{x}_k\| = 0. \tag{3.19}$$

Proof. By the instruction of the algorithm, we have $\mathbf{y}_k^1 = \mathbf{x}_k$ and for every $i = 2, \dots, n+1$,

$$\mathbf{y}_k^i = \mathbf{x}_k + \sum_{j=1}^{i-1} \alpha_k^j \mathbf{d}_{k+1}^j, \tag{3.20}$$

where $\mathbf{d}_{k+1}^i = \pm \mathbf{e}^i$, $i = 1, \dots, n$. Taking the limit in (3.20), for $k \to \infty, k \in \mathcal{K}_\rho^{\mathbf{x}}$, recalling that by Theorem 3.2.7, $\{\alpha_k^i\}_{k\in\mathcal{K}_\rho} \to 0$, and that, by assumption, $\lim\limits_{k\to\infty, k\in\mathcal{K}_\rho^{\mathbf{x}}} \mathbf{x}_k = \bar{\mathbf{x}}$, we have, for all $i = 1, \dots, n$,

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \mathbf{y}_k^i = \bar{\mathbf{x}}. \tag{3.21}$$

Let us now consider a direction $\mathbf{d}^i \in \mathcal{E} \cap T_{\mathcal{X}}(\bar{\mathbf{x}})$. By point (b) of Proposition 3.2.2, we know that for $k \in \mathcal{K}_\rho^{\mathbf{x}}$ sufficiently large $\mathbf{d}^i \in T_{\mathcal{X}}(\mathbf{y}_k^i)$, i.e. $\bar{\beta} > 0$ exists such that

$$\mathbf{y}_k^i + \beta \mathbf{d}^i \in \mathcal{X}, \qquad \forall\ \beta \in [0, \bar{\beta}].$$

By Theorem 3.2.7, we have that $\{\tilde{\alpha}_k^i\}_{k\in\mathcal{K}_\rho} \to 0$ and $\{\alpha_k^i\}_{k\in\mathcal{K}_\rho} \to 0$ (and $\alpha_k^i/\delta \to 0$). Hence, we have either

- $\alpha_k^i = 0$ which implies $0 < \tilde{\alpha}_k^i < \bar{\beta}$, for $k \in \mathcal{K}_\rho^{\mathbf{x}}$ sufficiently large, or
- $\alpha_k^i > 0$, which implies $\alpha_k^i < \bar{\beta}$ and $\frac{\alpha_k^i}{\delta} < \bar{\beta}$, for $k \in \mathcal{K}_\rho^{\mathbf{x}}$ sufficiently large.

If $\alpha_k^i = 0$, then

$$\begin{aligned} z(\mathbf{y}_k^i + \tilde{\alpha}_k^i \mathbf{d}_k^i; \rho_k) &> z(\mathbf{y}_k^i; \rho_k) - \gamma(\tilde{\alpha}_k^i)^2, \\ z(\mathbf{y}_k^i - \tilde{\alpha}_k^i \mathbf{d}_k^i; \rho_k) &> z(\mathbf{y}_k^i; \rho_k) - \gamma(\tilde{\alpha}_k^i)^2. \end{aligned} \tag{3.22}$$

If $\alpha_k^i > 0$, then we have either that an expansion has been performed along direction $-\mathbf{d}_k^i$ (in which case a failure has been obtained along $\mathbf{d}_k^i$); or an expansion has been performed along direction $\mathbf{d}_k^i$ (in which case direction $-\mathbf{d}_k^i$ has not been explicitly considered); in the latter case, we can still extract information along $-\mathbf{d}_k^i$ by setting $\tilde{\mathbf{y}}_k^i = \mathbf{y}_k^i + \tilde{\alpha}_k^i \mathbf{d}_k^i$ and obtain a suitable failure along $-\mathbf{d}_k^i$. Hence, we have either

$$\begin{aligned} z(\mathbf{y}_k^i + \tilde{\alpha}_k^i \mathbf{d}_k^i; \rho_k) &> z(\mathbf{y}_k^i; \rho_k) - \gamma(\tilde{\alpha}_k^i)^2, \\ z(\mathbf{y}_k^i - \frac{\alpha_k^i}{\delta} \mathbf{d}_k^i; \rho_k) &> z(\mathbf{y}_k^i; \rho_k) - \gamma(\frac{\alpha_k^i}{\delta})^2; \end{aligned} \tag{3.23}$$

in the former case, or

$$\begin{aligned} &z(\mathbf{y}_k^i + \frac{\alpha_k^i}{\delta} \mathbf{d}_k^i; \rho_k) > z(\mathbf{y}_k^i; \rho_k) - \gamma(\frac{\alpha_k^i}{\delta})^2, \\ &z(\tilde{y}_k^i - \tilde{\alpha}_k^i \mathbf{d}_k^i; \rho_k) \geq z(\tilde{\mathbf{y}}_k^i; \rho_k) + \gamma(\tilde{\alpha}_k^i)^2, \text{ with } \tilde{\mathbf{y}}_k^i = \mathbf{y}_k^i + \tilde{\alpha}_k^i \mathbf{d}_k^i, \end{aligned} \tag{3.24}$$

in the latter case.
When (3.22) holds, we choose $\mathbf{z}_k^i = \mathbf{y}_k^i$ and $\xi_k^i = \tilde{\alpha}_k^i$; when (3.23) holds, we choose $\mathbf{z}_k^i = \mathbf{y}_k^i$ and

$$\xi_k^i = \begin{cases} \tilde{\alpha}_k^i & \text{if } \mathbf{d}^i = \mathbf{d}_k^i \\ \frac{\alpha_k^i}{\delta} & \text{if } \mathbf{d}^i = -\mathbf{d}_k^i; \end{cases}$$

when (3.24) holds, we set

$$\begin{cases} \mathbf{z}_k^i = \mathbf{y}_k^i, \quad \xi_k^i = \frac{\alpha_k^i}{\delta} & \text{if } \mathbf{d}^i = \mathbf{d}_k^i \\ \mathbf{z}_k^i = \tilde{\mathbf{y}}_k^i, \quad \xi_k^i = \tilde{\alpha}_k^i & \text{if } \mathbf{d}^i = -\mathbf{d}_k^i. \end{cases}$$

Then, by (3.21) and recalling Theorem 3.2.7, we have that, for $k \in \mathcal{K}_\rho^{\mathbf{x}}$ sufficiently large, $\mathbf{z}_k^i$ and ξ_k^i, $i = 1, \dots, n$, are such that

$$\mathbf{z}_k^i + \xi_k^i \mathbf{d}^i \in \mathcal{X} \tag{3.25}$$

$$z(\mathbf{z}_k^i + \xi_k^i \mathbf{d}^i; \rho_k) > z(\mathbf{z}_k^i; \rho_k) - o(\xi_k^i); \tag{3.26}$$

furthermore,

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \xi_k^i = 0 \tag{3.27}$$

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \|\mathbf{z}_k^i - \mathbf{x}_k\| = 0. \tag{3.28}$$

We also recall that, given the definition of the penalty-barrier function $z(\cdot)$ and by Assumption 3.1 (i.e. $\mathbf{x}_0 \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$), for all k we have

$$\mathbf{z}_k^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g \ .$$

Now, let us assume there exists $\hat{t}_k^i \in (0, 1]$ such that $\mathbf{z}_k^i + \hat{t}_k^i \xi_k^i \mathbf{d}^i \notin \mathcal{X} \cap \overset{\circ}{\Omega}_g$. Then, by (3.25) and the compactness of $\mathcal{X}$

$$\mathbf{z}_k^i + \hat{t}_k^i \xi_k^i \mathbf{d}^i \notin \overset{\circ}{\Omega}_g, \quad \text{i.e. } g_\ell(\mathbf{z}_k^i + \hat{t}_k^i \xi_k^i \mathbf{d}^i) > 0, \text{ for some } \ell = 1, \dots, m.$$

Since $\mathbf{z}_k^i \in \overset{\circ}{\Omega}_g$, using the continuity assumption on the constraints, there exists a constant $\check{t}_k^i \in (0, \hat{t}_k^i]$ such that:

$$\max_{\ell=1,\dots,m} g_\ell(\mathbf{z}_k^i + \check{t}_k^i \xi_k^i \mathbf{d}^i) = 0 \text{ and } \mathbf{z}_k^i + t \xi_k^i \mathbf{d}^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g \ \text{ for all } t \in [0, \check{t}_k^i).$$

By the definition of $z(\cdot)$:

$$z(\mathbf{y}; \rho_k) \text{ is continuous } \forall \mathbf{y} \in [\mathbf{z}_k^i, \mathbf{z}_k^i + \check{t}_k^i \xi_k^i \mathbf{d}^i) \text{ and } \lim_{t \to \check{t}_k^i} z(\mathbf{z}_k^i + t \xi_k^i \mathbf{d}^i; \rho_k) = +\infty,$$

thus, a constant $t_k^{*,i} \in (0, \check{t}_k^i)$ exists such that:

$$z(\mathbf{z}_k^i + t_k^{*,i} \xi_k^i \mathbf{d}^i; \rho_k) > z(\mathbf{z}_k^i; \rho_k) - o(t_k^{*,i} \xi_k^i)$$

$$\mathbf{z}_k^i + t \xi_k^i \mathbf{d}^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g \quad \forall t \in [0, t_k^{*,i}].$$

By renaming $t_k^{*,i} \xi_k^i$ again ξ_k^i we have

$$z(\mathbf{z}_k^i + \xi_k^i \mathbf{d}^i; \rho_k) > z(\mathbf{z}_k^i; \rho_k) - o(\xi_k^i) \tag{3.29}$$

$$\mathbf{z}_k^i + t \xi_k^i \mathbf{d}^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g \quad \forall t \in [0, 1]. \tag{3.30}$$

Then, (3.27), (3.28), (3.29) and (3.30) finally conclude the proof. □

Finally, we can state the main result concerning convergence of the proposed `LOG-DFL` algorithm to stationary points of the original problem (P1).

Theorem 3.2.9. *Let* $\{\mathbf{x}_k\}$ *be the sequence generated by Algorithm* `LOG-DFL`. *Let* $\mathcal{K}_\rho$ *be the set of indices defined in (3.15). Assume that every limit point of the sequence* $\{\mathbf{x}_k\}_{\mathcal{K}_\rho}$ *satisfies the EMFCQ; then, every limit point* $\bar{x}$ *of the subsequence* $\{\mathbf{x}_k\}_{\mathcal{K}_\rho}$ *is a stationary point of problem* (P2).

Proof. Since $\{\mathbf{x}_k\}_{\mathcal{K}_\rho} \subseteq \mathcal{X}$ and $\mathcal{X}$ is a compact set, the subsequence $\{\mathbf{x}_k\}_{\mathcal{K}_\rho}$ admits limit points. Let us consider one such limit point $\bar{\mathbf{x}}$, i.e. an index set $\mathcal{K}_\rho^{\mathbf{x}} \subseteq \mathcal{K}_\rho$ exists such that

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x}}}} \mathbf{x}_k = \bar{\mathbf{x}}.$$

Let us denote $\bar{T} = \mathcal{E} \cap T_{\mathcal{X}}(\bar{\mathbf{x}})$. Recalling Proposition 3.2.8 we have that (3.16), (3.17), (3.18) and (3.19) hold.

Then, for all $k \in \mathcal{K}_\rho^{\mathbf{x}}$ sufficiently large and for all $\mathbf{d}^i \in \bar{T}$:

i) $\mathbf{y}_k^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$

ii) $\mathbf{y}_k^i + t\xi_k^i \mathbf{d}^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$ for all $t \in [0, 1]$

iii) the following inequality holds

$$z(\mathbf{y}_k^i + \xi_k^i \mathbf{d}^i; \rho_k) > z(\mathbf{y}_k^i; \rho_k) - o(\xi_k^i). \tag{3.31}$$

By applying the mean-value theorem to (3.31), we can write

$$-o(\xi_k^i) < z(\mathbf{y}_k^i + \xi_k^i \mathbf{d}^i; \rho_k) - z(\mathbf{y}_k^i; \rho_k) = \xi_k^i \nabla z(\mathbf{u}_k^i; \rho_k)^T \mathbf{d}^i \qquad \forall \mathbf{d}^i \in \bar{T}, \tag{3.32}$$

where $\mathbf{u}_k^i = \mathbf{y}_k^i + t_k^i \xi_k^i \mathbf{d}^i$, with $t_k^i \in (0, 1)$. By recalling (3.16), $\mathbf{u}_k^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$. Thus, we have

$$-\frac{o(\xi_k^i)}{\xi_k^i} < \nabla z(\mathbf{u}_k^i; \rho_k)^T \mathbf{d}^i \qquad \forall \mathbf{d}^i \in \bar{T}.$$

By considering the expression of $z(\mathbf{x}; \rho)$, we can write

$$\begin{aligned}\nabla z(\mathbf{u}_k^i; \rho_k)^T \mathbf{d}^i = & \Bigg(\nabla f(\mathbf{u}_k^i) + \sum_{\ell=1}^{m} \frac{\rho_k}{-g_\ell(\mathbf{u}_k^i)} \nabla g_\ell(\mathbf{u}_k^i) \\ & + \sum_{j=1}^{q} \frac{\nu}{\rho_k} |h_j(\mathbf{u}_k^i)|^{\nu-1} \nabla h_j(\mathbf{u}_k^i)\Bigg)^T \mathbf{d}^i > -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall \mathbf{d}^i \in \bar{T}.\end{aligned} \tag{3.33}$$

Recalling that $\mathbf{u}_k^i = \mathbf{y}_k^i + t_k^i \xi_k^i \mathbf{d}^i$, with $t_k^i \in (0, 1)$, we have that, for all i such that $\mathbf{d}^i \in \bar{T}$,

$$\lim_{\substack{k\to\infty,\\ k\in\bar{\mathcal{K}}}} \mathbf{u}_k^i = \bar{\mathbf{x}}. \tag{3.34}$$

Now, let us define the following approximations of the multipliers.

$$\begin{aligned}\lambda_\ell(\mathbf{x}; \rho) &= \frac{\rho}{-g_\ell(\mathbf{x})}, \quad \text{for all } \ell = 1, \dots, m \\ \mu_j(\mathbf{x}; \rho) &= \frac{\nu}{\rho} |h_j(\mathbf{x})|^{\nu-1}, \quad \text{for all } j = 1, \dots, q.\end{aligned}$$

It is possible to show that the sequences $\{\lambda_\ell(\mathbf{x}_k;\rho_k)\}_{\mathcal{K}_\rho^{\mathbf{x}}}$, $\ell = 1,\dots,m$, and $\{\mu_j(x_k;\rho_k)\}_{\mathcal{K}_\rho^{\mathbf{x}}}$, $j = 1,\dots,q$ are bounded. The proof of this property is rather technical and, to simplify the exposition, the result is reported in Theorem 3.4.7 in Section 3.4.

Then there exists a subset of indices $\mathcal{K}_\rho^{\mathbf{x},\lambda} \subseteq \mathcal{K}_\rho^{\mathbf{x}}$, such that

$$\lim_{\substack{k\to\infty,\\ k\in\mathcal{K}_\rho^{\mathbf{x},\lambda}}} \lambda_\ell(\mathbf{x}_k;\rho_k) = \bar\lambda_\ell \geq 0, \qquad \ell = 1,\dots,m, \tag{3.35}$$

$$\lim_{k\to\infty, k\in\mathcal{K}_\rho^{\mathbf{x},\lambda}} \mu_j(x_k;\rho_k) = \bar\mu_j \geq 0, \qquad j = 1,\dots,q, \tag{3.36}$$

where $\bar\lambda_\ell = 0$ for $\ell \notin \mathcal{I}^+(\bar{\mathbf{x}})$.

Since $\mathbf{y}_k^i \in \mathcal{X} \cap \overset{\circ}{\Omega}_g$ and, by continuity of $g_\ell(\mathbf{x})$, for all $\ell = 1,\dots,m$, set Ω_g is closed, any accumulation point of the sequence $\{\mathbf{y}_k^i\}$ belongs to $\mathcal{X} \cap \Omega_g$. We consider now the sequence of positive penalty parameters ρ_k. By Theorem 3.2.7, we have that:

$$\lim_{k\to\infty} \rho_k = 0.$$

Then, multiplying (3.33) by ρ_k and taking the limit for $k \to \infty$, $k \in \mathcal{K}_\rho^{\mathbf{x},\lambda}$, recalling (3.35) and the continuity assumptions, we have:

$$\left(\sum_{j=1}^{p} \nu |h_j(\bar{\mathbf{x}})|^{\nu-1} \nabla h_j(\bar{\mathbf{x}})\right)^T \mathbf{d}^i \geq 0 \qquad \forall \mathbf{d}^i \in \bar{T}. \tag{3.37}$$

Since $\bar{\mathbf{x}}$ satisfies EMFCQ, by (3.2), it results:

$$h_j(\bar{\mathbf{x}}) = 0 \qquad \forall j = 1,\dots,p.$$

Therefore the point $\bar{\mathbf{x}} \in \mathcal{F}$, i.e. it is feasible for problem (P2).

By simple manipulations, (3.33) can be rewritten as

$$\begin{aligned}
&\Bigg(\nabla f(\mathbf{u}_k^i) + \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{u}_k^i)\lambda_\ell(\mathbf{x}_k;\rho_k) \\
&+ \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{u}_k^i)\left(\lambda_\ell(\mathbf{u}_k^i;\rho_k) - \lambda_\ell(\mathbf{x}_k;\rho_k)\right) + \sum_{j=1}^{q} \nabla h_j(\mathbf{u}_k^i)\mu_j(\mathbf{x}_k;\rho_k) \\
&+ \sum_{j=1}^{q} \nabla h_j(\mathbf{u}_k^i)\left(\mu_j(\mathbf{u}_k^i;\rho_k) - \mu_j(\mathbf{x}_k;\rho_k)\right)\Bigg)^T \mathbf{d}^i > -\frac{o(\xi_k^i)}{\xi_k^i} \qquad \forall i : \mathbf{d}^i \in \bar{T}.
\end{aligned} \tag{3.38}$$

Taking the limits for $k \to \infty$ and $k \in \mathcal{K}_\rho^{\mathbf{x},\lambda}$ in relation (3.38) and recalling (3.98) and (3.99) from the proof of Proposition 3.4.7 previously invoked, we obtain

$$\left(\nabla f(\bar{\mathbf{x}}) + \sum_{\ell=1}^{m} \nabla g_\ell(\bar{\mathbf{x}})\bar\lambda_\ell + \sum_{j=1}^{q} \nabla h_j(\bar{\mathbf{x}})\bar\mu_j\right)^T \mathbf{d}^i \geq 0 \qquad \forall i : \mathbf{d}^i \in \bar{T}.$$

Recalling that $\bar{T} = \mathcal{E} \cap T_{\mathcal{X}}(\bar{\mathbf{x}})$, from point (a) of Proposition 3.2.2 we get

$$\nabla L(\bar{\mathbf{x}}, \bar{\boldsymbol\lambda}, \bar{\boldsymbol\mu})^T d \geq 0 \qquad \forall \mathbf{d} \in T_{\mathcal{X}}(\bar{x}),$$

which concludes the proof. □

To conclude the section, we highlight the different reasoning at the basis of Proposition 3.2.8 and Theorem 3.2.9 with respect to Proposition 4 and Theorem 1 in [73]. In particular, Proposition 3.2.8 shows that algorithm `LOG-DFL` is able to produce suitable points and step-sizes, i.e. that satisfy some specific conditions. Then, Theorem 3.2.9 builds on top of the latter results and shows convergence to stationary points. On the contrary, [73, Proposition 4] gives minimal conditions that should be guaranteed in order to get convergence to stationary points. Then [73, Theorem 1] basically shows that the algorithm is able to guarantee the convergence conditions thus showing convergence to stationary points.

3.2.4 Numerical experiments

In this section we report the numerical performance of the proposed log-barrier derivative-free Algorithm `LOG-DFL` on a set of test problems chosen from a well-known collection.

Test problem collection

In this subsection we report the set of constrained test problems selected from the CUTEst collection [49]. In particular, we selected all the problems with $n \leq 50$ variables and having at least one inequality constraint for which the provided initial point is strictly feasible. The first numerical experimentation considers the test problems discarding all the violated constraints. We then propose an alternative version of the penalty function which is able to handle violated inequalities, therefore such constraints will be taken into account. The collection provides a total of 96 problems.

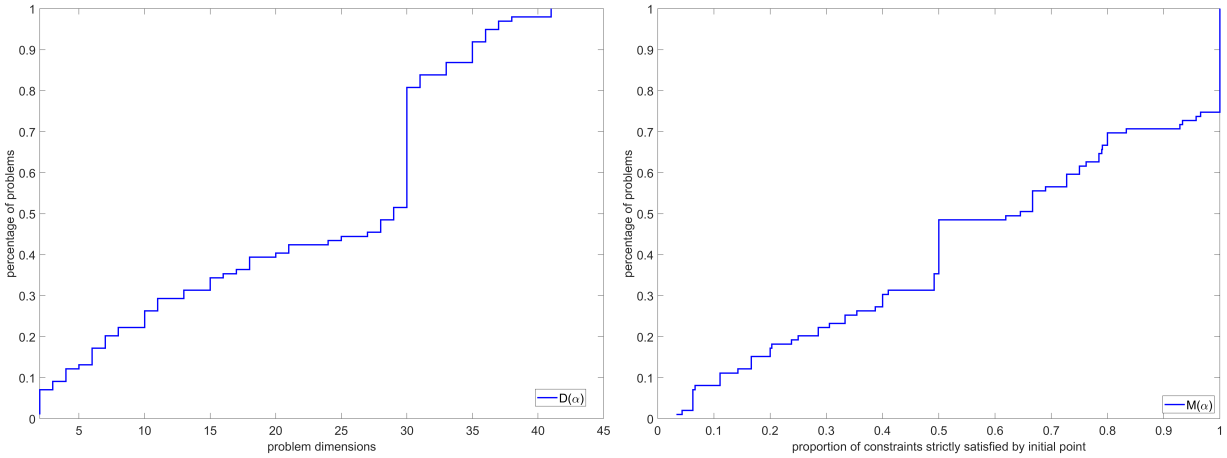

Figure 3.1. Cumulative distributions, respectively, of the number of variables and of the proportion of strictly satisfied constraints with respect to the total number of constraints.

In figure 3.1 we report the cumulative distributions, respectively, of the number of variables and of the proportion of strictly satisfied constraints with respect to the total number of constraints, i.e.

$$\begin{aligned} D(\alpha) &= \frac{1}{N} \left|\{p \in \mathcal{P} : \; n_p \leq \alpha\}\right| \\ M(\alpha) &= \frac{1}{N} \left|\left\{p \in \mathcal{P} : \; \frac{\bar{m}_p}{m_p} \leq \alpha\right\}\right| \end{aligned}$$

where

- $\mathcal{P}$ is the set of problems;
- $N = |\mathcal{P}|$;
- n_p is the number of variables of problem $p \in \mathcal{P}$;
- m_p is the number of constraints of problem $p \in \mathcal{P}$;
- $\bar{m}_p$ is the number of strictly satisfied inequality constraints at the initial point for problem $p \in \mathcal{P}$.

Problem	n_p	m_p	$\bar{m}_p$	meq_p
ANTWERP	27	10	2	8
DEMBO7	16	21	16	0
ERRINBAR	18	9	1	8
HS117	15	5	5	0
HS118	15	29	28	0
LAUNCH	25	29	20	9
LOADBAL	31	31	20	11
MAKELA4	21	40	20	0
MESH	33	48	17	24
OPTPRLOC	30	30	28	0
RES	20	14	2	12
SYNTHES2	11	15	1	1
SYNTHES3	17	23	1	2
TENBARS1	18	9	1	8
TENBARS4	18	9	1	8
TRUSPYR1	11	4	1	3
TRUSPYR2	11	11	8	3
HS12	2	1	1	0
HS13	2	1	1	0
HS16	2	2	2	0
HS19	2	2	1	0
HS20	2	3	3	0
HS21	2	1	1	0
HS23	2	5	4	0
HS30	3	1	1	0
HS43	4	3	3	0
HS65	3	1	1	0
HS74	4	5	2	3
HS75	4	5	2	3
HS83	5	6	5	0
HS95	6	4	3	0
HS96	6	4	3	0
HS97	6	4	2	0
HS98	6	4	2	0
HS100	7	4	4	0
HS101	7	6	2	0
HS104	8	6	3	0
HS105	8	1	1	0
HS113	10	8	8	0
HS114	10	11	8	3
HS116	13	15	10	0
S365	7	5	2	0
ALLINQP	24	18	9	3
BLOCKQP1	35	16	1	15
BLOCKQP2	35	16	1	15
BLOCKQP3	35	16	1	15
BLOCKQP4	35	16	1	15
BLOCKQP5	35	16	1	15
CAMSHAPE	30	94	90	0
CAR2	21	21	5	16
CHARDIS1	28	14	13	0
EG3	31	90	60	1
GAUSSELM	29	36	11	14
GPP	30	58	58	0
HADAMARD	37	93	36	21
HANGING	15	12	8	0
JANNSON3	30	3	2	1
JANNSON4	30	2	2	0
KISSING	37	78	32	12
KISSING1	33	144	113	0
KISSING2	33	144	113	0
LIPPERT1	41	80	64	16
LIPPERT2	41	80	64	16
LUKVLI1	30	28	28	0
LUKVLI10	30	28	14	0
LUKVLI11	30	18	3	0
LUKVLI12	30	21	6	0
LUKVLI13	30	18	3	0
LUKVLI14	30	18	18	0
LUKVLI15	30	21	7	0
LUKVLI16	30	21	13	0
LUKVLI17	30	21	21	0
LUKVLI18	30	21	21	0
LUKVLI2	30	14	7	0
LUKVLI3	30	2	2	0
LUKVLI4	30	14	4	0
LUKVLI6	31	15	15	0
LUKVLI8	30	28	14	0
LUKVLI9	30	6	6	0
MANNE	29	20	10	0
MOSARQP1	36	10	10	0
MOSARQP2	36	10	10	0
NGONE	29	134	106	0
NUFFIELD	38	138	28	0
OPTMASS	36	30	6	24
POLYGON	28	119	94	0
POWELL20	30	30	15	0
READING4	30	60	30	0
SINROSNB	30	58	29	0
SVANBERG	30	30	30	0
VANDERM1	30	59	29	30
VANDERM2	30	59	29	30
VANDERM3	30	59	29	30
VANDERM4	30	59	29	30
YAO	30	30	1	0
ZIGZAG	28	30	5	20

Table 3.1. Test set selected from the CUTEst collection. Parameters n_p, m_p, $\bar{m}_p$, and meq_p denote, respectively, the number of variables, of inequality constraints, of inequality constraints treated by the logarithmic barrier, and of equality constraints for the given problem. The problems in boldface are used for the second numerical comparison detailed in Section 3.3.5.

Implementation details

The proposed method has been implemented in Python 3.9, and all the experiments have been conducted adopting the following choices:

- *Exponent parameter for exterior penalty*

  $\nu = 2$.

- *Parameters introduced in the* `LOG-DFL` *algorithm scheme*

  $\gamma = 10^{-4}$, $\beta = 1 + 10^{-10}$, $\tilde{\alpha}_0^i = 1.0$.

- *Penalty parameters initialization*

  In order to properly balance the different ways the two terms of penalization impact the merit function we define two different parameters $\rho_0^{\texttt{in}}$, $\rho_0^{\texttt{ex}}$. We can write the new penalized function:

  $$z(\mathbf{x}, \rho_k) = f(\mathbf{x}) - \rho_k^{\texttt{in}} \sum_{\ell=1}^{m} \log\left[-g_\ell(x)\right] + \frac{1}{\rho_k^{\texttt{ex}}} \sum_{j=1}^{q} |h_j(\mathbf{x})|^{\nu}. \tag{3.39}$$

  We initialize the two penalty parameters with the following rules:

  $$\rho_0^{\texttt{in}} = 10^{-1}$$

  $$\rho_0^{\texttt{ex}} = \min\left\{10^{-3}, \frac{1}{max\{|f(\mathbf{x}_0)|, 10^{-10}\}}\right\}$$

- *Penalty parameter updating criterion*

  According to our theoretical results and those in [73], we use two different updating criteria for the penalty $\rho_k^{\texttt{ex}}$ and barrier $\rho_k^{\texttt{in}}$ parameter:

  $$\max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} \le (\rho_k^{\texttt{ex}})^{\beta}, \tag{3.40}$$

  $$\max_{i=1,2,\dots,n} \{\tilde{\alpha}_k^i, \alpha_k^i\} \le \min\left\{(\rho_k^{\texttt{in}})^{p}, (g_{\min})_k^2\right\}, \tag{3.41}$$

  where we remind that $(g_{\min})_k$ is the minimum absolute value for the non-relaxable constraints at iteration k.

  Concerning the parameter θ, we split it into two different parameters: $\theta^{\texttt{in}} = 0.35$ and $\theta^{\texttt{ex}} = 10^{-2}$. When (3.41) is satisfied the algorithm performs the following update:

  $$\rho_{k+1}^{\texttt{in}} = \theta^{in} \rho_k^{\texttt{in}}.$$

  When (3.40) and (3.41) are both satisfied the algorithm performs the following update:

  $$\rho_{k+1}^{\texttt{ex}} = \theta^{\texttt{ex}} \rho_k^{\texttt{ex}}.$$

  Note that the rule (3.41) is more restrictive than (3.40), and $\theta^{\texttt{ex}} < \theta^{\texttt{in}}$. Hence, the new updating mechanism is able to guarantee that both the parameters, $\rho_k^{\texttt{ex}}$ and $\rho_k^{\texttt{in}}$, go to zero with the same speed and that there is an infinite subsequence of iterations where they are both updated.

- *Stopping criterion*

  We stop the algorithm whenever $\max_{i=1,2,\dots,n}\{\tilde{\alpha}_k^i, \alpha_k^i\} \leq 10^{-8}$.

The `LOG-DFL` algorithm is freely available for download through the DFL library as package `LOGDFL` at the URL https://github.com/DerivativeFreeLibrary/
For comparison we used the state-of-the-art MADS algorithm implemented in the well-known `NOMAD` package (version 4) [9]

Note that `NOMAD` does not allow for explicit handling of equality constraints. For this reason we had to split the equality constraints into two inequalities.

Furthermore the resulting inequalities are treated by the progressive approach specifying the `PB` type of constraint. Namely, for each equality constraint h_j, $j = 1, \dots, q$, we define

$$\begin{aligned} g_{m+j}(\mathbf{x}) &= h_j(\mathbf{x}), \quad \texttt{PB} \\ g_{m+q+j}(\mathbf{x}) &= -h_j(\mathbf{x}), \quad \texttt{PB}. \end{aligned}$$

We also implemented a basic mechanism to force `NOMAD` to stay feasible with respect to the inequality constraints. In particular, we forced `NOMAD` to handle those constraints with an extreme barrier approach. Furthermore, since `LOG-DFL` and `NOMAD` use different strategies to handle constraints, it might not be fair to compare them using the same precision on the stopping criterion. Thus, in order to be sure that `NOMAD` would exploit the evaluations budget as much as possible, we set the stopping criterion relative to the mesh size δ to 10^{-12}. It has to be pointed out that, during this phase of the experiments, we are considering the inequality constraints to be non-relaxable, however NOMAD builds surrogate models of the constraints and the objective using also information from points which violate the inequalities. Since NOMAD is able to handle infinite values, we limited the information on those points by manually setting infinite values for the objective function, still leaving the information on feasibility and violation available.

Performance and data profiles

Numerical experiments will be analyzed using performance [41] and data [78] profiles. To provide a brief overview of these tools, consider a set of solvers $\mathcal{S}$ and a set of problems $\mathcal{P}$. Let $t_{p,s}$ represent the number of function evaluations required by solver s to satisfy the convergence test adopted for problem p.

For an accuracy $\tau = 10^{-k}$, where $k \in \{1, 3, 5\}$, we adopted the convergence test:

$$f_M - f(\mathbf{x}) \geq (1 - \tau)(f_M - f_L), \tag{3.42}$$

where f_M represents the objective function value of the worst feasible point determined by all solvers for problem p, and f_L is the best objective function value obtained by all solvers, corresponding to a feasible point of problem p.

The convergence test given by (3.42) requires a significant reduction in the objective function value by comparison with the worst feasible point f_M. We assign an infinite value to the objective function at points that violate the feasibility conditions, defined by $c(\mathbf{x}) > 10^{-4}$, where

$$c(\mathbf{x}) = \sum_{i=1}^{m} \max\{0, g_i(\mathbf{x})\} + \sum_{j=1}^{q} |h_j(\mathbf{x})|.$$

The performance of solver $s \in \mathcal{S}$ is measured by the fraction of problems in which the performance ratio is at most α, given by:

$$\rho_s(\alpha) = \frac{1}{|P|} \left| \left\{ p \in P \mid \frac{t_{p,s}}{\min \left\{ t_{p,s'} : s' \in S \right\}} \leq \alpha \right\} \right|.$$

A performance profile provides an overview of how well a solver performs across a set of optimization problems. Particularly relevant is the value $\rho_s(1)$, that reflects the efficiency of the solver, i.e., the percentage of problems for which the algorithm performs the best. Robustness, as the percentage of problems that the algorithm is able to solve, can be perceived for high values of α.

Data profiles focus on the behavior of the algorithm during the optimization process. A data profile measures the percentage of problems that can be solved (given the tolerance τ) with κ estimates of simplex gradients and is defined by:

$$d_s(\kappa) = \frac{1}{|P|} \left| \left\{ p \in P \mid t_{p,s} \leq \kappa \left(n_p + 1 \right) \right\} \right|,$$

where n_p represents the dimension of problem p.

Results and comparison

We divide the subsection in three parts. We first introduce *accelerating* strategies that helped to improve the performance of `LOG-DFL` and show the related results. Then we compare the proposed algorithm with `NOMAD`, first on the collection of problems presented before, and afterwards considering the same collection of problems where the inequalities violated at the starting point are not discarded and treated by means of an exterior penalty of the same expression of the penalization for the equality constraints. All the comparisons are done considering a budget of $100 \times (n+1)$ function evaluations.

`Accelerating strategies for LOG-DFL`

In the following, we first describe two heuristics that helped us improve the basic version of the `LOG-DFL` algorithm. Note that there is no theoretical evidence the *accelerating* strategies are able to improve the performances of the algorithm, but they do not affect the theoretical convergence properties of the method. Indeed, both strategies can be considered as **Step 3** of Algorithm `LOG-DFL`.

1. First (**S1**), at each iteration the method attempts to use one more direction to try to exploit the overall movement of the current iteration itself. The direction is computed as $(\mathbf{d}_{n+1})_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. We then perform a further exploration by means of a suitable modification of the expansion step procedure, which makes sure the generated points are feasible with respect to the bounds, i.e. $\mathcal{X}$. It is a naive approach for a minimal exploitation of the information gathered at each iteration. We believe the strategy could be further improved by computing $(\mathbf{d}_{n+1})_k$ building surrogate models of the merit function, since the structure of the method can guarantee a good geometry of the points generated at each iteration, which are the ones that would be interpolated.

2. Second (**S2**), since the merit function changes its values when using a different penalty parameter, it is reasonable to think that, when the parameter is modified, the current iterate might not be the best among the previously visited ones. Thus, at all the iterations the penalty

parameter is changed, we select from the history of the algorithm the point with the best value of the current merit function and use it for the following iteration.

`LOG-DFL` algorithm with accelerating strategies is freely available for download through the DFL library as package `LOGDFL` at the URL https://github.com/DerivativeFreeLibrary/

The comparison between our original algorithm `LOG-DFL` and the improved method (which we call `X-LOG-DFL`) are reported in Figure 3.2 and 3.3. As we can see, `X-LOG-DFL` is significantly better than the original version both in terms of efficiency and robustness. One can see how both accelerating strategies allow for improvement with respect to the original method, especially for higher level of precision.

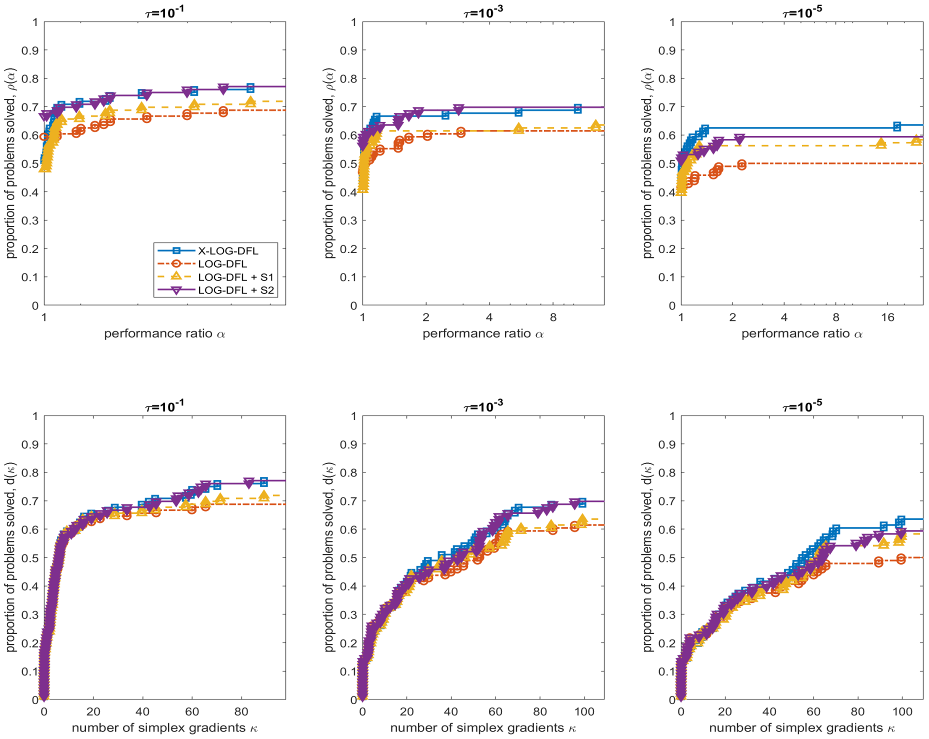

Figure 3.2. Performance and data profiles for the comparison between `LOG-DFL` and the accelerating strategies among the whole collection of test problems.

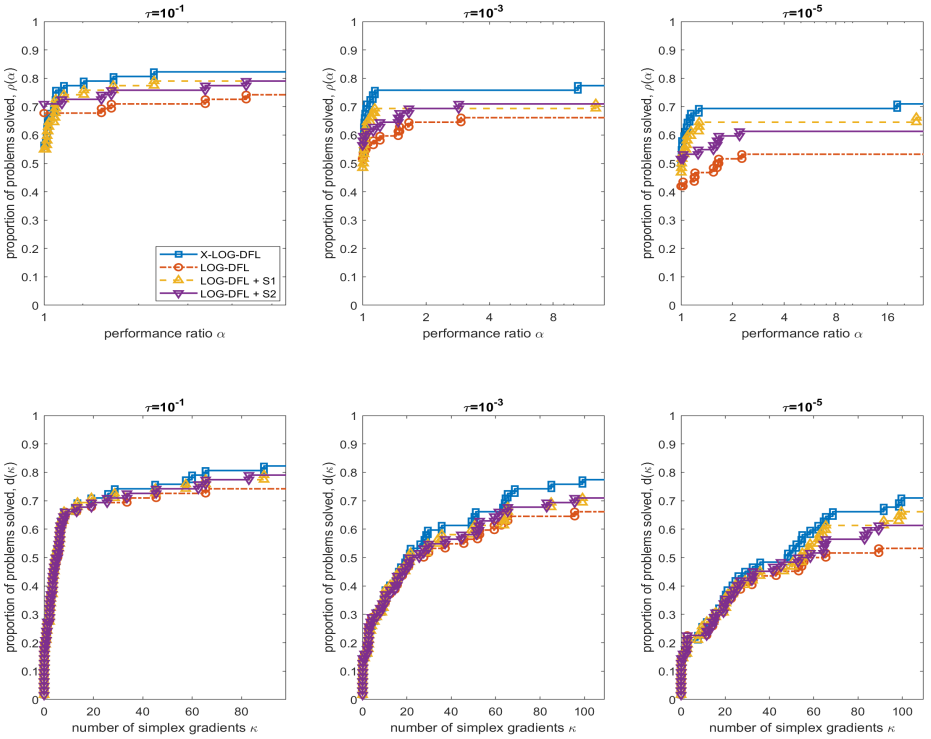

Figure 3.3. Performance and data profiles for the comparison between `LOG-DFL` and the accelerating strategies among the problems with only inequality constraints.

Comparison with strictly feasible inequalities

In Figures 3.4 and 3.5, we report the comparison between `X-LOG-DFL`, `LOG-DFL` and `NOMAD`. As one can see, considering either the results in Figure 3.4, where some problems have equality constraints, i.e. `NOMAD` has no theoretical guarantees, or the results in 3.5, where only problems with inequality constraints are considered, both `LOG-DFL` and `X-LOG-DFL` are competitive against `NOMAD`. In particular, both algorithms perform better than `NOMAD` for low precision levels, and `X-LOG-DFL` is performing better also on higher precision levels, where, from the performance profiles, it seems to be able to reach better quality solutions. Note that while `X-LOG-DFL` is using no information at all for points that are not feasible with respect to the inequality constraints, `NOMAD`, for the same points, exploits the values of the constraints to build surrogate models, i.e., `X-LOG-DFL` is performing better using less information than the competitor.

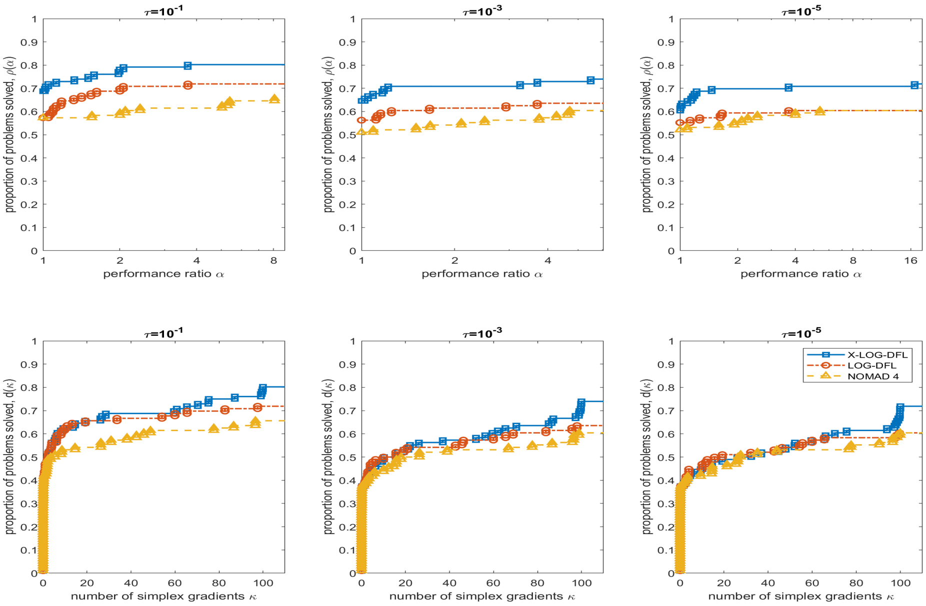

Figure 3.4. Performance and data profiles for the comparison between `X-LOG-DFL`, `LOG-DFL` and `NOMAD` on the entire collection of test problems.

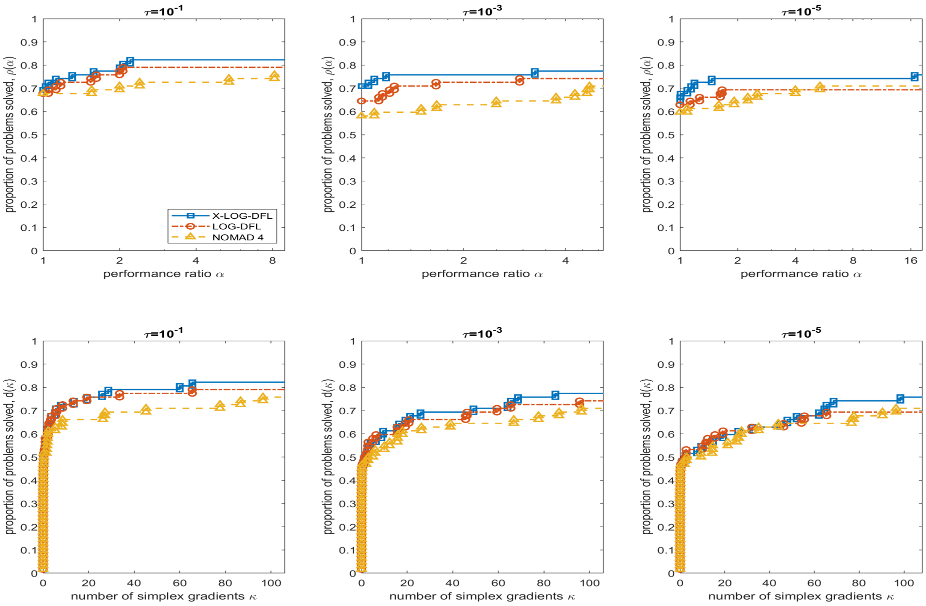

Figure 3.5. Performance and data profiles for the comparison between `X-LOG-DFL`, `LOG-DFL`, and `NOMAD` on the subset of test problems with only inequality constraints.

Comparison with violated inequality constraints

To conclude the section, we further investigate the behaviour of the competing algorithms introducing a new term to the penalty function, with the purpose of dealing with problems where inequality constraints might be violated at the starting point. The modifications of the method affect only the part of the penalty parameters initialization. We computed the values of the constraints at the starting point $\mathbf{x}_0$ and we defined two sets of indices:

$$\mathcal{G}^{\texttt{log}} := \{\ell : g_\ell(\mathbf{x}_0) < 0\},$$

$$\mathcal{G}^{\texttt{ext}} := \{\ell : g_\ell(\mathbf{x}_0) \geq 0\}.$$

We set a logarithmic penalty for strictly feasible constraints and exterior penalty for active and infeasible ones. We define two parameters $\rho_0^{\texttt{log}}$, $\rho_0^{\texttt{ext}}$, and we initialize them with the rules we defined in the previous section. We can write the new penalized function:

$$\begin{aligned} z(\mathbf{x}, \rho_k) \quad &= \quad f(\mathbf{x}) - \rho_k^{\texttt{log}} \sum_{\ell \in \mathcal{G}^{\texttt{log}}} \log\left[-g_\ell(\mathbf{x})\right] + \\ &\qquad \frac{1}{\rho_k^{\texttt{ext}}} \left(\sum_{\ell \in \mathcal{G}^{\texttt{ext}}} [g_\ell^+(\mathbf{x})]^\nu + \sum_{j=1}^{q} |h_j(\mathbf{x})|^\nu \right), \end{aligned} \tag{3.43}$$

where $g_\ell^+(\mathbf{x}) = \max\{g_\ell(\mathbf{x}), 0\}$.

As one can see in (3.43), we are now using a hybrid approach, where some inequality constraints are handled by interior penalty and some are handled by exterior penalty. In fact, equality constraints are treated by splitting them into two inequality constraints, which will be managed by exterior penalty. Furthermore, we add a new strategy with the aim of exploiting the efficiency of the logarithmic barrier: when an initially infeasible inequality (which is penalized by using the exterior penalty approach) becomes feasible, we change the penalization method adopted for that particular constraint, thus switching to an interior point penalization. For this part of the experiments we consider the inequalities to be relaxable, so we let `NOMAD` run in default settings, i.e. using PB (progressive barrier) and exploiting all the information of the problem, with the only specification that the algorithm will keep running as long as the mesh size parameter δ is greater than 10^{-12}. Note that `X-LOG-DFL` still does not exploit information on points where the inequalities g_ℓ with $\ell \in \mathcal{G}^{\texttt{log}}$ are violated. The results are reported in Figures 3.6, 3.7 and 3.8. On the problems without equality constraints, Figure 3.8, the two algorithms show a similar performance. The results appear to be similar also for the problems where both solvers find a feasible solution, Figure 3.7, with `X-LOG-DFL` being faster when considering the two highest levels of precision. It is worth noticing that, in the latter case, even though both solvers find a feasible solution, the two algorithms do not solve the whole subset of test problems. That means that the algorithms are reaching different solutions for the different problems. As it might seem an obvious conclusion, it is also a sign that further numerical investigation is needed in order to understand which kind of problems each algorithm performs the better on. In conclusion, when considering the whole collection of problems, Figure 3.6, `X-LOG-DFL` performs better than `NOMAD`.

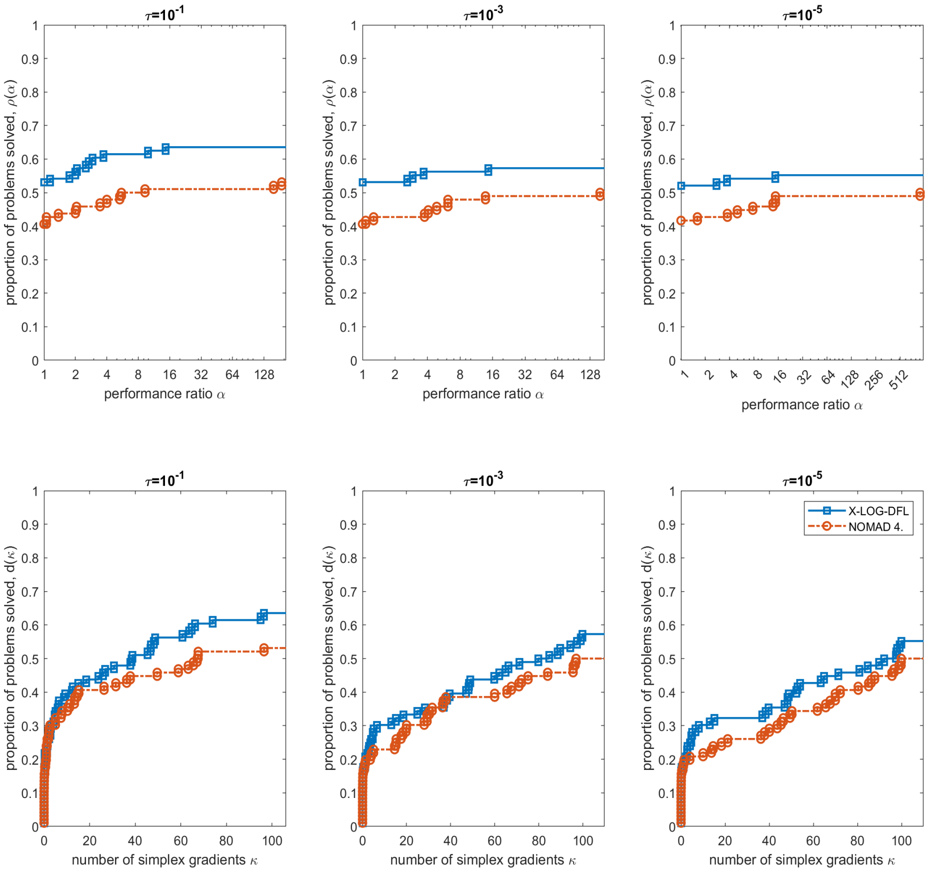

Figure 3.6. Performance and data profiles for the comparison between `X-LOG-DFL`, `NOMAD` for the whole collection of test problems considering also violated inequalities.

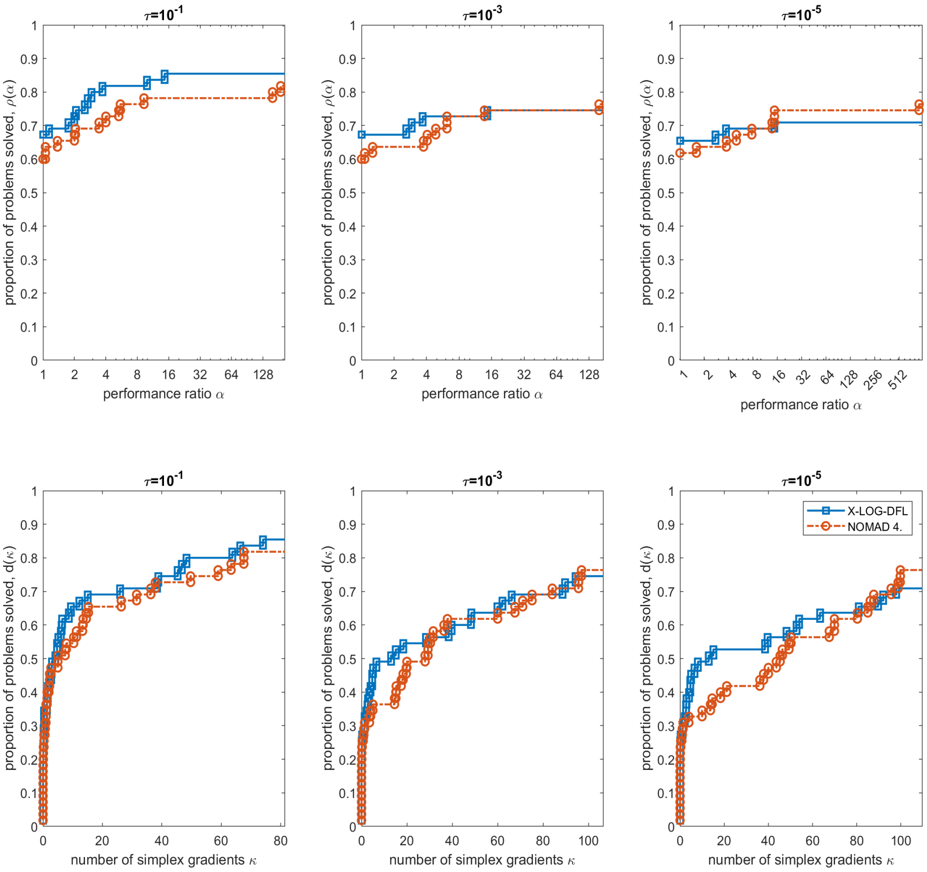

Figure 3.7. Performance and data profiles for the comparison between `X-LOG-DFL` and `NOMAD` on the subset of problems where both solvers find feasible solutions.

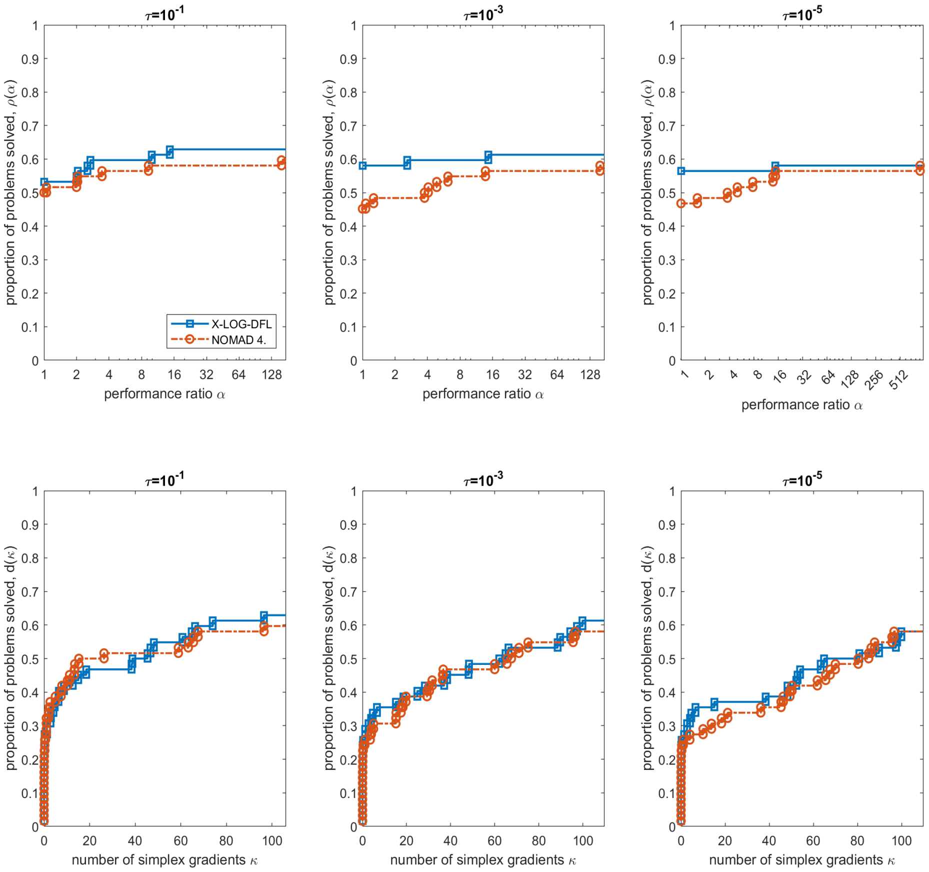

Figure 3.8. Performance and data profiles for the comparison between `X-LOG-DFL` and `NOMAD` on the subset of problems without equality constraints.

3.3 LOG-DS

In this section we propose a DS algorithm which exploits the MPB strategy to solve problem (P2). The structure of this section is similar to that of the previous one. In Section 3.2.1 the assumptions used are detailed along with some preliminary results. In Section 3.2.2, we introduce `LOG-DS` and we explain its main features. In Section 3.2.3 we prove some properties of `LOG-DS` and the convergence to KKT-stationary points. Finally, In Section 3.2.4 describe how we have practically implemented the algorithm and we present the comparison against state-of-the-art solvers.

3.3.1 Assumptions and preliminary results

We consider (P2). Following the idea of (3.43) in the previous section, we partition the inequality constraints into two sets, namely $\mathcal{G}^{\texttt{log}}$ and $\mathcal{G}^{\texttt{ext}}$, such that $\mathcal{G}^{\texttt{log}} \cap \mathcal{G}^{\texttt{ext}} = \emptyset$ and $\mathcal{G}^{\texttt{log}} \cup \mathcal{G}^{\texttt{ext}} = \{1, \dots, m\}$. We denote the sets defined by the inequality constraints as $\Omega_{\texttt{log}} = \{\mathbf{x} \in \mathbb{R}^n \mid g_\ell(\mathbf{x}) \le 0, \ell \in \mathcal{G}^{\texttt{log}}\}$ and $\Omega_{\texttt{ext}} = \{\mathbf{x} \in \mathbb{R}^n \mid g_\ell(\mathbf{x}) \le 0, \ell \in \mathcal{G}^{\texttt{ext}}\}$, respectively. The first corresponds to the inequality constraints to be addressed with a logarithmic barrier approach, while the constraints in the second are handled using an infeasible penalization term. Additionally, Ω_h corresponds to the set defined by the equality constraints, and $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \le \mathbf{b}\}$, with $\mathbf{A} \in \mathbb{R}^{p \times n}$ and $\mathbf{b} \in \mathbb{R}^p$, is a polyhedron, defining a set that we assume to be compact. Therefore, the feasible region $\mathcal{F}$, which is assumed to be nonempty, is given by

$$\mathcal{F} = \mathcal{X} \cap \Omega_{\texttt{log}} \cap \Omega_{\texttt{ext}} \cap \Omega_h \ne \emptyset,$$

and it is also a compact set. Differently to the method proposed in Section 3.2, where the goal was to propose an efficient method to treat unrelaxable constraints, the aim of this section is to show the efficiency of the mixed penalty strategy in more general settings. Therefore, here we are not assuming the inequality constraints to be unrelaxable, even though the method is able to deal with the presence of unrelaxable constraints.

Throughout this section, we require the following assumption to hold true.

Assumption 3.2.

- *The set* $\mathcal{X}$ *is compact.*
- *A point* $\mathbf{x}_0 \in \mathcal{X} \cap \mathring{\Omega}_{\texttt{log}}$ *is known.*

The set $\mathcal{X}$ includes the bounds on the variables, thus assuming its compactness is viable. Furthermore, given any initial solution $\mathbf{x}_0$, it is possible to define the sets $\mathcal{G}^{\texttt{log}}$ and $\mathcal{G}^{\texttt{ext}}$ so that the assumption is satisfied, as it has been done in the previous section. Note that in case $g_\ell(\mathbf{x}_0) \ge 0$ for all $\ell = 1, \dots, m$, then we have $\mathcal{G}^{\texttt{log}} = \emptyset$ and $\Omega_{\texttt{log}} = \mathbb{R}^n$.

In order to solve Problem (P2), the following merit function is considered:

$$z^+(\mathbf{x}; \rho) = f(\mathbf{x}) - \rho \sum_{\ell \in \mathcal{G}^{\texttt{log}}} \log(-g_\ell(\mathbf{x})) + \frac{1}{\rho^{\nu-1}} \left(\sum_{\ell \in \mathcal{G}^{\texttt{ext}}} (\max\{g_\ell(\mathbf{x}), 0\})^\nu + \sum_{j=1}^{q} |h_j(\mathbf{x})|^\nu \right), \tag{3.44}$$

where $\rho > 0$ and $\nu \in (1, 2]$. Therefore, $z^+(\mathbf{x}; \rho) = +\infty$, for all $\mathbf{x} \in \mathcal{X}$ such that $g_\ell(\mathbf{x}) \not< 0$, for $\ell \in \mathcal{G}^{\texttt{log}}$.

The following problem

$$\begin{aligned} &\min \ z^+(\mathbf{x};\rho) \\ &\text{s.t. } \mathbf{x} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}} \end{aligned} \tag{C2}$$

will be considered at each iteration, with an adequate strategy for updating the parameter $\rho > 0$. We assume that a point $\mathbf{x}_0 \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$ is known, so that the set $\mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$ is nonempty.

Lemma 3.3.1. *Let $\rho > 0$, $\nu \in (1, 2]$, and $\alpha \in \mathbb{R}$ be given parameters. If $\mathcal{X} \cap \Omega_{\mathtt{log}}$ is compact, then,*

$$L(\alpha) = \{\mathbf{x} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}} : \ z^+(\mathbf{x};\rho) \le \alpha\}$$

is compact.

Proof. The set $L(\alpha)$ is bounded since, by definition, it is a subset of $\mathcal{X} \cap \Omega_{\mathtt{log}}$ which is compact. It remains to prove that $L(\alpha)$ is closed. To this end, we will show that for any sequence $\{\mathbf{x}_k\} \subset L(\alpha)$ such that $\lim\limits_{k\to+\infty} \mathbf{x}_k = \bar{\mathbf{x}}$, it results $\bar{\mathbf{x}} \in L(\alpha)$.

Since $\mathbf{x}_k \in L(\alpha)$, for all k, we have

$$z^+(\mathbf{x}_k;\rho) \le \alpha.$$

Taking the limit for $k \to +\infty$ in the above relation we get

$$\lim_{k\to+\infty} z^+(\mathbf{x}_k;\rho) \le \alpha. \tag{3.45}$$

Then, $\bar{\mathbf{x}} \notin \partial\mathring{\Omega}_{\mathtt{log}}$, otherwise we would get $\lim_{k\to+\infty} z^+(\mathbf{x}_k;\rho) = +\infty > \alpha$. Thus, $\bar{\mathbf{x}} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$. We know that $z^+(\mathbf{x};\rho)$ is continuous on $\mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$. Thus, by (3.45), we have

$$\lim_{k\to+\infty} z^+(\mathbf{x}_k;\rho) = z^+(\bar{\mathbf{x}};\rho) \le \alpha$$

which means that $\bar{\mathbf{x}} \in L(\alpha)$. That concludes the proof. □

One might notice that nonlinear constraints still appear, due to the presence of the inequalities in $\mathcal{G}^{\mathtt{log}}$ which still need to be satisfied. Nevertheless, considering the properties of the merit function and the structure of the proposed scheme, it will be clear that any point $\mathbf{x} \notin \mathring{\Omega}_{\mathtt{log}}$ will be rejected. That is, Problem C2 can be reformulated considering only linear constraints, i.e.,

$$\begin{aligned} &\min \ z^+(\mathbf{x};\rho) \\ &\text{s.t. } \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{P4}$$

In order to deal with the linear constraints, one has to be able to capture the geometry of $\mathcal{X}$ around a point $\mathbf{x} \in \mathcal{X}$. In the following we provide the proper definitions of the tangent cone with respect to $\mathcal{X}$.

Definition 3.3.2 (Active constraints and tangent related sets)**.** *For every $\mathbf{x} \in \mathcal{X}$, i.e., such that $\mathbf{A}\mathbf{x} \le \mathbf{b}$:*

$$\mathcal{I}_{\mathcal{X}}(\mathbf{x}) = \{i \mid \mathbf{a}_i^\top \mathbf{x} = b_i\} \quad \textit{(set of indices of active constraints)}$$
$$T_{\mathcal{X}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \mathbf{a}_i^\top \mathbf{d} \le 0, \ i \in \mathcal{I}_{\mathcal{X}}(\mathbf{x})\} \quad \textit{(tangent cone at } \mathbf{x}\textit{)}$$

Given a point $\mathbf{x} \in \mathcal{X}$ (possibly not belonging to the boundary of $\mathcal{X}$), it is possible to approximate the previous sets by the following ones, depending on a parameter $\varepsilon > 0$. Note that this definition is an extension of the ϵ-tangent cones defined in Section 2.4.2 for the bound constraints.

Definition 3.3.3 (ε-Active constraints and tangent related sets)**.**

$$\mathcal{I}_{\mathcal{X}}(\mathbf{x},\varepsilon) = \{i \mid \mathbf{a}_i^\top \mathbf{x} \geq b_i - \varepsilon\} \quad \textit{(set of indices of } \varepsilon\textit{-active constraints)}$$
$$T_{\mathcal{X}}(\mathbf{x},\varepsilon) = \{\mathbf{d} \in \mathbb{R}^n \mid \mathbf{a}_i^\top \mathbf{d} \leq 0,\ i \in \mathcal{I}_{\mathcal{X}}(\mathbf{x},\varepsilon)\} \quad \textit{(}\varepsilon\textit{-tangent cone at } \mathbf{x}\textit{)}$$

A relation between the two groups of sets introduced in Definition 3.3.2 and Definition 3.3.3 has been established in [76]. We recall the result in the following proposition.

Proposition 3.3.4. *Let $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ be a sequence of points in $\mathcal{X}$, converging to $\mathbf{x}^* \in \mathcal{X}$. Then, there exists an $\varepsilon^* > 0$ (depending only on $\mathbf{x}^*$) such that for any $\varepsilon \in (0,\varepsilon^*]$ there exists $k_\varepsilon \in \mathbb{N}$ such that*

$$\mathcal{I}_{\mathcal{X}}(\mathbf{x}^*) = \mathcal{I}_{\mathcal{X}}(\mathbf{x}_k,\varepsilon)$$
$$T_{\mathcal{X}}(\mathbf{x}^*) = T_{\mathcal{X}}(\mathbf{x}_k,\varepsilon)$$

for all $k \geq k_\varepsilon$.

Finally, we provide a definition that will be useful to specify the requests on the directions used by the algorithm in the following section.

Definition 3.3.5 (ε-conformed to $\mathcal{X}$)**.** *Let $\mathbf{x} \in \mathcal{X}$. A set of directions $\mathcal{P}$ is said to be $\bar{\varepsilon}$-conformed to $\mathcal{X}$ in $\mathbf{x}$ if for some $\bar{\varepsilon} > 0$*

$$\operatorname{cone}(\mathcal{P} \cap T_{\mathcal{X}}(\mathbf{x},\varepsilon)) = T_{\mathcal{X}}(\mathbf{x},\varepsilon), \quad \forall\ \varepsilon \in (0,\bar{\varepsilon}].$$

3.3.2 LOG-DS Algorithm

Let us introduce the structure of the `LOG-DS` algorithm, based on the sequential inexact minimization of linearly constrained subproblems, i.e. (P4), differing from each other in the penalty parameter ρ.

```
LOG-DS
Data. $\mathbf{x}_0 \in \mathcal{X}$ such that $g_\ell(\mathbf{x}_0) < 0$ for all $\ell \in \mathcal{G}^{\log}$, $\alpha_0 > 0$, $\rho_0 > 0$, $\nu \in (1,2]$, $\theta_\alpha, \theta_\rho \in (0,1)$, $\phi \geq 1$, and $\beta > 1$.
For $k = 0,1,2,\dots$ do
    Step 1. (Search Step, optional)
            If $\mathbf{z}_k \in \mathcal{X}$ can be computed such that $z^+(\mathbf{z}_k;\rho_k) \leq z^+(\mathbf{x}_k;\rho_k) - \xi(\alpha_k)$,
            Then set $\mathbf{x}_{k+1} = \mathbf{z}_k$, $\alpha_{k+1} = \phi\alpha_k$, and go to Step 3.
    Step 2. (Poll Step)
            Select a set of directions $\mathcal{P}_k$
            If $\mathbf{x}_k + \alpha_k\mathbf{t} \in \mathcal{X}$ and $z^+(\mathbf{x}_k + \alpha_k\mathbf{t};\rho_k) \leq z^+(\mathbf{x}_k;\rho_k) - \xi(\alpha_k)$ for some $\mathbf{t} \in \mathcal{P}_k$,
            Then set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k^i$ and $\alpha_{k+1} = \phi\alpha_k$.
            Else set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and $\alpha_{k+1} = \theta_\alpha\alpha_k$.
    Step 3. Set $(g_{\min})_k = \min_{\ell\in\mathcal{G}^{\log}} \{|g_\ell(\mathbf{x}_{k+1})|\}$
            If $\alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\min})_k^2\}$ and $\alpha_{k+1} < \alpha_k$,
            Then set $\rho_{k+1} = \theta_\rho\rho_k$.
            Else set $\rho_{k+1} = \rho_k$.
Endfor
```

In the method detailed above, the acceptance of new points relies on the notion of sufficient decrease. The next definition adjusts the concept of forcing function (see [60]) and it is used to define the sufficient decrease condition required to accept new points in `LOG-DS`.

Definition 3.3.6. *Let $\xi : [0, +\infty) \to [0, +\infty)$ be a continuous and nondecreasing function. We say that ξ is a forcing function if:*

- $\xi(t)/t \to 0$ *when* $t \downarrow 0$*;*
- *if* $\xi(t) \to 0$ *then* $t \to 0$.

Following the general structure proposed by Audet and Dennis [7] for generalized pattern search, each iteration of `LOG-DS` is organized into two main steps, plus an additional one related to the novel approach:

- Step 1 and Step 2, namely the Search Step and the Poll Step, which are part of the basic structure of a generalized pattern search method, here applied to the solution of (P4);
- Step 3, the penalty parameter updating step, which is the original feature of the proposed approach.

The search step is very flexible, not even requiring the projection of the generated points in some type of implicit mesh, since a sufficient decrease condition is used for the acceptance of new iterates. As detailed in Section 3.3.4, the original `SID-PSM` algorithm uses quadratic interpolation models, which are minimized to generate new trial points. The latter approach is adapted into `LOG-DS` as described in Section 4.1, but since it is not relevant for establishing convergence properties, for now it is omitted.

We are now in position of specifying the requests on the sets of directions, $\mathcal{P}_k$, used by the algorithm (see [72, Assumption 2]). Note that in Chapter 1, when describing the MADS algorithm, the set $\mathcal{P}_k$ is defined as a subset of the points lying on the mesh within the frame. Here, using a sufficient decrease condition, we are not using the mesh, and the directions are directly generated as a subset of $\mathcal{D}$. The set $\mathcal{P}_k$, then, denotes the directions used during the Poll Step.

Assumption 3.3. *Let $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ be a sequence of points in X. The sequence $\{\mathcal{P}_k\}$ of poll directions satisfies:*

$$\mathcal{P}_k = \{\mathbf{d}_k^i \mid \|\mathbf{d}_k^i\| = 1, i = 1, \ldots, |\mathcal{P}_k|\}$$

and for some $\bar{\varepsilon} > 0$,

$$\operatorname{cone}(\mathcal{P}_k \cap T_{\mathcal{X}}(\mathbf{x}_k, \varepsilon)) = T_{\mathcal{X}}(\mathbf{x}_k, \varepsilon), \quad \forall\, \varepsilon \in (0, \bar{\varepsilon}].$$

Furthermore, $\mathcal{D} = \bigcup_{k=0}^{+\infty} \mathcal{P}_k$ is a finite set, and $|\mathcal{P}_k|$ is uniformly bounded.

Strategies to conform the poll directions to the geometry of $\mathcal{X}$, satisfying Assumption 3.3, can be found in [1, 61, 66].

One problem still to be addressed is related to the fact that the merit function is defined and differentiable only in the interior of the feasible region, i.e. $\mathring{\Omega}_{\mathcal{G}^{\log}}$. An analysis based on differentiability assumptions seems not to be applicable. It turns out that the structure of the merit function can be exploited to extract information that allows the infeasible points to be treated as feasible failures, as it will be stated in Proposition 3.3.9, similarly to as it is proved in Proposition 3.2.8 for `LOG-DFL`.

3.3.3 Convergence Analysis

In order to prove the global convergence of the method, we will establish that the sequences of step-sizes and penalty parameters converge to zero. Initially, we derive an auxiliary result analyzing the behavior of the algorithm for fixed values of the penalty parameter.

Lemma 3.3.7. *Let $\{\rho_k\}$ and $\{\alpha_k\}$ be the sequences of penalty parameters and step-sizes, respectively, generated by algorithm* `LOG-DS`. *Assume that*

$$\lim_{k\to+\infty} \rho_k = \bar{\rho} > 0 \tag{3.46}$$

Then,

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

Proof. From the updating rule of the penalty parameter, i.e. $\rho_{k+1} = \theta_\rho \rho_k$, we have that $\{\rho_k\}$ is a monotone non-increasing sequence. Furthermore, if ρ_k is updated infinitely many times we would have $\bar{\rho} = 0$. Hence, we have that $\rho_{k+1} = \rho_k = \bar{\rho}$ for all k sufficiently large. Let us split the iteration sequence into the following two sets

$$\mathcal{K}_s = \{k : \mathbf{x}_{k+1} \neq \mathbf{x}_k\},$$
$$\mathcal{K}_u = \{k : \mathbf{x}_{k+1} = \mathbf{x}_k\}.$$

At every iteration k of the algorithm, for k sufficiently large, we have either $z^+(\mathbf{x}_{k+1};\bar{\rho}) = z^+(\mathbf{x}_k;\bar{\rho})$ (when $k \in \mathcal{K}_u$) or $z^+(\mathbf{x}_{k+1};\bar{\rho}) \le z^+(\mathbf{x}_k;\bar{\rho}) - \xi(\alpha_k)$ (when $k \in \mathcal{K}_s$). Hence, the sequence of function values $\{z^+(\mathbf{x}_k;\bar{\rho})\}$ is monotonically nonincreasing. By Lemma 3.3.1, $z^+(\mathbf{x};\bar{\rho})$ has compact level sets, thus it is bounded from below. Hence,

$$\lim_{k\to+\infty} z^+(\mathbf{x}_k;\bar{\rho}) = \bar{z}. \tag{3.47}$$

If $\mathcal{K}_s$ is infinite, from (3.47) we get

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_s}} \xi(\alpha_k) = 0.$$

Recalling Definition 3.3.6, we get

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_s}} \alpha_k = 0. \tag{3.48}$$

If $\mathcal{K}_u$ is infinite, for every $k \in \mathcal{K}_u$, let us define m_k to be the largest index such that $m_k \in \mathcal{K}_s$ and $m_k < k$ (the result is immediate if $\mathcal{K}_s$ is empty). Then, we can write

$$\alpha_k = \alpha_{m_k} \phi \theta_\alpha^{k-m_k-1}.$$

When $k \to +\infty$, $k \in \mathcal{K}_u$, we have that either $m_k \to +\infty$ as well (when $\mathcal{K}_s$ is infinite) or $k - m_k - 1 \to +\infty$ (when $\mathcal{K}_s$ is finite). Thus, by (3.48) and the fact that $\theta_\alpha \in (0,1)$, we have

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_u}} \alpha_k = \lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_u}} \alpha_{m_k} \phi \theta_\alpha^{k-m_k-1} = 0. \tag{3.49}$$

The proof is concluded considering (3.48) and (3.49). □

Lemma 3.3.7 is used to show that the sequence of penalty parameters converges to zero, which is required to ensure that in the limit the algorithm can solve the original problem. As a consequence, we are able to establish that the sequence of step-sizes also converges to zero in the general case. Since it is well-known that the step-size is related to some measures of stationarity of the problem [60], that property is also relevant.

Theorem 3.3.8. *Let* $\{\rho_k\}$ *and* $\{\alpha_k\}$ *be the sequences of penalty parameters and step-sizes generated by* `LOG-DS`*. Let* $\mathcal{K}_\rho = \{k \in \mathbb{N} : \rho_{k+1} < \rho_k\}$*. Then,*

$$\lim_{k \to +\infty} \rho_k = 0, \tag{3.50}$$

$$\lim_{k \to +\infty, k \in \mathcal{K}_\rho} \alpha_k = 0. \tag{3.51}$$

Proof. We first prove (3.50). The algorithmic structure implies that $\{\rho_k\}_{k \in \mathbb{N}}$ is a monotone nonincreasing sequence of positive numbers. Hence, we have that

$$\lim_{k \to +\infty} \rho_k = \bar{\rho} \geq 0.$$

By contradiction, let us assume that $\bar{\rho} > 0$. If this is the case, there must exist an integer $\bar{k}$ such that

$$\rho_{k+1} = \rho_k = \bar{\rho} > 0, \quad \forall k \geq \bar{k}.$$

Again, the instructions of the algorithm imply that, for all $k \geq \bar{k}$,

$$\alpha_{k+1} > \min\{\bar{\rho}^\beta, (g_{\min})_k^2\} \text{ or } \alpha_{k+1} \geq \alpha_k$$

and

$$z^+(\mathbf{x}_{k+1}; \bar{\rho}) \leq z^+(\mathbf{x}_k; \bar{\rho}) \leq z^+(\mathbf{x}_{\bar{k}}; \bar{\rho}) = \bar{z} < +\infty. \tag{3.52}$$

Since $\rho_k = \bar{\rho}$, for all $k \geq \bar{k}$, by Lemma 3.3.7 we have $\lim\limits_{k \to +\infty} \alpha_k = 0$. Then, there must exist an infinite index set of unsuccessful iterations $\mathcal{K}_u$ such that $\alpha_{k+1} < \alpha_k$, for all $k \in \mathcal{K}_u$. Hence, for every $k \in \mathcal{K}_u$, $k \geq \bar{k}$, we also have that

$$\alpha_{k+1} > \min\{\bar{\rho}^\beta, (g_{\min})_k^2\} \tag{3.53}$$

(otherwise, the algorithm would update ρ). Taking the limit in relation (3.53), we obtain

$$\lim_{\substack{k \to +\infty, \\ k \in \mathcal{K}_u}} (g_{\min})_k = 0.$$

Thus, we have

$$\lim_{\substack{k \to +\infty, \\ k \in \mathcal{K}_u}} z^+(\mathbf{x}_k; \bar{\rho}) = +\infty.$$

However, this limit is in contradiction with (3.52), thus proving (3.50).

Now, we prove (3.51). Since ρ_k is updated at each $k \in \mathcal{K}_\rho$, then it follows that $\mathcal{K}_\rho \subseteq \mathcal{K}_u$, and $\alpha_{k+1} = \theta_\alpha \alpha_k$. Thus, by the penalty-barrier parameter updating criterion, for all $k \in \mathcal{K}_\rho$

$$\theta_\alpha \bar{\alpha} \leq \alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\min})_k^2\} \leq \rho_k^\beta,$$

which, since $\rho_k \to 0$, implies $\{\alpha_k\}_{k \in \mathcal{K}_\rho} \to 0$, thus concluding the proof. □

Let us define the index set

$$\mathcal{K}_\rho = \{k \in \mathbb{N} : \rho_{k+1} < \rho_k\}, \tag{3.54}$$

that is, the set of iteration indices where the penalty parameter is updated. Note that, by the instructions of the algorithm, $\alpha_{k+1} < \alpha_k$ for all $k \in \mathcal{K}_\rho$, i.e., every iteration $k \in \mathcal{K}_\rho$ is an unsuccessful iteration. Recall that, by Theorem 3.3.8, $\mathcal{K}_\rho$ is an infinite index set.

Before proving the main convergence result, since the merit function is not defined in $\mathbb{R}^n \backslash \mathring{\Omega}_{\mathcal{G}}^{\mathtt{log}}$ and the trial points corresponding to failures might not be feasible, we present the following proposition that allows us to use the mean value theorem in the proof of Theorem 3.3.10.

Proposition 3.3.9. *Given any* $\bar\rho > 0$, $\mathbf{x} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$, $\mathbf{d} \in \mathbb{R}^n$, *and* $\bar\alpha \in \mathbb{R}_+$ *such that* $\mathbf{x} + \bar\alpha\mathbf{d} \in \mathcal{X}$ *and*

$$z^+(\mathbf{x} + \bar\alpha\mathbf{d}; \bar\rho) > z^+(\mathbf{x}; \bar\rho) - \xi(\bar\alpha),$$

there exists $\hat\alpha < \bar\alpha$ *such that:*

$$\begin{aligned} &\mathbf{x} + \alpha\mathbf{d} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}} \;\; \textit{for all} \;\; \alpha \in (0, \hat\alpha], \\ &z^+(\mathbf{x} + \hat\alpha\mathbf{d}; \bar\rho) > z^+(\mathbf{x}; \bar\rho) - \xi(\hat\alpha). \end{aligned}$$

Proof. By definition $\mathcal{X}$ is convex, then $\mathbf{x} + \alpha\mathbf{d} \in \mathcal{X}$ for all $\alpha \in [0, \bar\alpha]$, thus if $\mathbf{x} + \alpha\mathbf{d} \in \mathcal{X} \cap \mathring{\Omega}_{\mathtt{log}}$ for all $\alpha \in (0, \bar\alpha]$, setting $\hat\alpha = \bar\alpha$ the proposition holds.
Let us assume $\mathbf{x} + \alpha\mathbf{d} \notin \mathring{\Omega}_{\mathtt{log}}$ for some $\alpha \in (0, \bar\alpha]$. Since $\mathbf{x} \in \mathring{\Omega}_{\mathtt{log}}$, so that $g_\ell(\mathbf{x}) < 0$ for all $\ell \in \mathcal{G}^{\mathtt{log}}$, and $\mathbf{x} + \bar\alpha\mathbf{d} \notin \mathring{\Omega}_{\mathtt{log}}$, by continuity of $g_\ell(\cdot)$, there exists a scalar $\tilde\alpha < \bar\alpha$ such that

$$\mathbf{x} + \alpha\mathbf{d} \in \mathring{\Omega}_{\mathtt{log}} \text{ for all } \alpha \in [0, \tilde\alpha),$$

$$\mathbf{x} + \tilde\alpha\mathbf{d} \in \partial\mathring{\Omega}_{\mathtt{log}}.$$

Recall that, by definition of $z^+(\mathbf{x}; \rho)$, for all $\mathbf{y} \in \partial\mathring{\Omega}_{\mathtt{log}}$, it results

$$\lim_{\substack{\mathbf{x} \to \mathbf{y}, \\ \mathbf{x} \in \mathring{\Omega}_{\mathtt{log}}}} z^+(\mathbf{x}; \bar\rho) = +\infty.$$

Thus, there exists $\hat\alpha \in (0, \tilde\alpha)$, sufficiently close to $\tilde\alpha$, such that $\mathbf{x} + \hat\alpha\mathbf{d} \in \mathring{\Omega}_{\mathtt{log}}$ and $z^+(\mathbf{x}; \bar\rho) < z^+(\mathbf{x} + \hat\alpha\mathbf{d}; \bar\rho) + \xi(\hat\alpha)$, which concludes the proof. ☐

We are now in conditions of stating the main convergence result. In the following we use the extended Mangasarian-Fromovitz constraint qualification (EMFCQ) as defined in Definition 3.2.3, and the same necessary optimality conditions introduced in Proposition 3.2.4.

Theorem 3.3.10. *Let* $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ *be the sequence of iterates generated by* `LOG-DS` *and recall definition* (3.54) *of* $\mathcal{K}_\rho$. *Assume that the sequences of sets of directions* $\{\mathcal{P}_k\}_{k\in\mathbb{N}}$, *used by the algorithm, satisfy Assumption 3.3 and define* $\mathcal{J}_k = \{i \in \{1, 2, \dots, |\mathcal{P}_k|\} : \mathbf{d}_k^i \in \mathcal{P}_k \cap T_{\mathcal{X}}(\mathbf{x}_k; \varepsilon)\}$, *with* $\varepsilon \in (0, \min\{\bar\varepsilon, \varepsilon^*\}]$ *where* ε^* *and* $\bar\varepsilon$ *are the constants appearing in Proposition 3.3.4 and Assumption 3.3, respectively. Then, any limit point of* $\{\mathbf{x}_k\}_{k\in\mathcal{K}_\rho}$ *that satisfies the EMFCQ is a stationary point of Problem* (P2).

Proof. First note that, by Theorem 3.3.8, we have

$$\lim_{\substack{k \to +\infty \\ k \in \mathcal{K}_\rho}} \rho_k = 0,$$

$$\lim_{\substack{k \to +\infty \\ k \in \mathcal{K}_\rho}} \alpha_k = 0.$$

Now, let $\mathbf{x}^*$ be any limit point of $\{\mathbf{x}_k\}_{k\in\mathcal{K}_\rho}$. Then, there exists a set $\mathcal{K}_\rho^{\mathtt{x}} \subseteq \mathcal{K}_\rho$ such that

$$\begin{aligned}
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x}}}} \rho_k = 0,\\
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x}}}} \alpha_k = 0,\\
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x}}}} \mathbf{x}_k = \mathbf{x}^*,
\end{aligned}$$

with $\alpha_{k+1} < \alpha_k$, for all $k \in \mathcal{K}_\rho^{\mathtt{x}}$. Recall that $\mathcal{P}_k = \{\mathbf{d}_k^1, \mathbf{d}_k^2, \dots, \mathbf{d}_k^{r_k}\}$. Then, for all $k \in \mathcal{K}_\rho^{\mathtt{x}}$ sufficiently large, we know that $\mathbf{x}_k + \alpha_k \mathbf{d}_k^i \in \mathcal{X}$ for all $i \in \mathcal{J}_k$. For every $i \in \mathcal{J}_k$, if $\mathbf{x}_k + \alpha_k \mathbf{d}_k^i \in \mathring{\Omega}_{\mathtt{log}}$, by the instructions of the algorithm we have

$$z^+(\mathbf{x}_k + \alpha_k \mathbf{d}_k^i; \rho_k) > z^+(\mathbf{x}_k; \rho_k) - \xi(\alpha_k).$$

Otherwise, i.e. when $\mathbf{x}_k + \alpha_k \mathbf{d}_k^i \notin \mathring{\Omega}_{\mathtt{log}}$, Proposition 3.3.9 allows us to ensure the existence of a scalar $\hat{\alpha}_k^i \le \alpha_k$ such that

$$z^+(\mathbf{x}_k + \hat{\alpha}_k^i \mathbf{d}_k^i; \rho_k) > z^+(\mathbf{x}_k; \rho_k) - \xi(\hat{\alpha}_k^i). \tag{3.55}$$

Applying the mean value theorem to (3.55), we can write

$$-\xi(\hat{\alpha}_k^i) \le z^+(\mathbf{x}_k + \hat{\alpha}_k^i \mathbf{d}_k^i; \rho_k) - z^+(\mathbf{x}_k; \rho_k) = \hat{\alpha}_k^i \nabla z^+(\mathbf{y}_k^i; \rho_k)^\top \mathbf{d}_k^i, \tag{3.56}$$

for all $k \in \mathcal{K}_\rho^{\mathtt{x}}$ sufficiently large and all $i \in \mathcal{J}_k$, where $\mathbf{y}_k^i = \mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i$, with $t_k^i \in (0,1)$. Thus, we have

$$\nabla z^+(\mathbf{y}_k^i; \rho_k)^\top \mathbf{d}_k^i \ge -\frac{\xi(\hat{\alpha}_k^i)}{\hat{\alpha}_k^i}, \quad \forall i \in \mathcal{J}_k. \tag{3.57}$$

By considering the expression of $z^+(\mathbf{x}; \rho_k)$, we can write for $k \in \mathcal{K}_\rho^{\mathtt{x}}$ sufficiently large

$$\begin{aligned}
\nabla z^+(\mathbf{y}_k^i; \rho_k)^\top \mathbf{d}_k^i \;=\; &\left[\nabla f(\mathbf{y}_k^i) + \sum_{\ell\in\mathcal{G}^{\mathtt{log}}} \frac{\rho_k}{-g_\ell(\mathbf{y}_k^i)} \nabla g_\ell(\mathbf{y}_k^i) + \nu \left(\sum_{\ell\in\mathcal{G}^{\mathtt{ext}}} \left(\frac{\max\{g_\ell(\mathbf{y}_k^i), 0\}}{\rho_k} \right)^{\nu-1} \nabla g_\ell(\mathbf{y}_k^i) + \right.\right.\\
&\left.\left. \sum_{j=1}^{p} \left(\frac{|h_j(\mathbf{y}_k^i)|}{\rho_k} \right)^{\nu-1} \nabla h_j(\mathbf{y}_k^i) \right) \right]^\top \mathbf{d}_k^i \ge -\frac{\xi(\hat{\alpha}_k^i)}{\hat{\alpha}_k^i}, \quad \forall i \in \mathcal{J}_k.
\end{aligned} \tag{3.58}$$

By Assumption 3.3, we can extract a further subset of indices $\mathcal{K}_\rho^{\mathtt{x,d}} \subseteq \mathcal{K}_\rho^{\mathtt{x}}$ such that, $\alpha_{k+1} < \alpha_k$, for all $k \in \mathcal{K}_\rho^{\mathtt{x,d}}$ and

$$\begin{aligned}
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d}}}} \rho_k = 0\\
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d}}}} \alpha_k = 0,\\
&\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d}}}} \mathbf{x}_k = \mathbf{x}^*,\\
&\mathcal{J}_k = \bar{\mathcal{J}}, \quad \forall\, k \in \mathcal{K}_\rho^{\mathtt{x,d}},\\
&\mathbf{d}_k^i = \bar{\mathbf{d}}^i, \quad \forall\, i \in \bar{\mathcal{J}},\ k \in \mathcal{K}_\rho^{\mathtt{x,d}},
\end{aligned} \tag{3.59}$$

and $\mathcal{T}^* = \{\bar{\mathbf{d}}^i\}_{i\in\bar{\mathcal{J}}}$. When $k \in \mathcal{K}_\rho^{\mathtt{x,d}}$ is sufficiently large, for all $i \in \bar{\mathcal{J}}$, with $\mathbf{y}_k^i = \mathbf{x}_k + t_k^i \hat{\alpha}_k^i \bar{\mathbf{d}}^i$, and $t_k^i \in (0,1)$, since $\hat{\alpha}_k^i \le \alpha_k$, by Theorem 3.3.8, we have that, $\displaystyle\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d}}}} \mathbf{y}_k^i = \mathbf{x}^*$.

Let us define the following approximations to the Lagrange multipliers of each constraint:

- for $\ell = 1,\dots,m$ set $\lambda_\ell(\mathbf{x};\rho) = \begin{cases} \dfrac{\rho}{-g_\ell(\mathbf{x})}, & \text{if } \ell \in \mathcal{G}^{\mathtt{log}} \\ \nu\left(\dfrac{\max\{g_\ell(\mathbf{x}),0\}}{\rho}\right)^{\nu-1}, & \text{if } \ell \in \mathcal{G}^{\mathtt{ext}} \end{cases}$
- for $j = 1,\dots,p$ set $\mu_j(\mathbf{x};\rho) = \nu\left(\dfrac{|h_j(\mathbf{x})|}{\rho}\right)^{\nu-1}$.

The sequences $\{\lambda_\ell(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathtt{x}}}$ and $\{\mu_j(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathtt{x}}}$, are bounded (see Theorem 3.4.6 in Section 3.4). Thus, it is possible to consider $\mathcal{K}_\rho^{\mathtt{x,d},\lambda} \subseteq \mathcal{K}_\rho^{\mathtt{x,d}}$, such that

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d},\lambda}}} \lambda_\ell(\mathbf{x}_k;\rho_k) = \lambda_\ell^*, \qquad \ell = 1,\dots,m \tag{3.60}$$

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x,d},\lambda}}} \mu_j(\mathbf{x}_k;\rho_k) = \mu_j^*, \qquad j = 1,\dots,p \tag{3.61}$$

and define $\lambda_\ell^* = 0$ for $\ell \notin \mathcal{I}^+(\mathbf{x}^*)$.

Multiplying (3.58) by $\rho_k^{\nu-1}$ and taking the limit for $k \to +\infty, k \in \mathcal{K}_\rho^{\mathtt{x,d},\lambda}$, recalling $\nu \in (1,2]$, we have that $\rho_k^{\nu-1} \to 0$, so we obtain the following

$$\left(\sum_{\ell\in\mathcal{G}^{\mathtt{ext}}} \nu\max\{g_\ell(\mathbf{x}^*),0\}^{\nu-1}\nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p}\nu|h_j(\mathbf{x}^*)|^{\nu-1}\nabla h_j(\mathbf{x}^*)\right)^\top \bar{\mathbf{d}}^i \ge 0, \quad \forall\bar{\mathbf{d}}^i \in \mathcal{T}^*.$$

From Proposition 3.3.4 and Assumption 3.3, we know that there is $\varepsilon > 0$ such that for all $k \in \mathcal{K}_\rho^{\mathtt{x,d},\lambda}$ sufficiently large

$$T_{\mathcal{X}}(\mathbf{x}^*) = T_{\mathcal{X}}(\mathbf{x}_k;\varepsilon) = \mathrm{cone}(\mathcal{P}_k \cap T_{\mathcal{X}}(\mathbf{x}_k;\varepsilon)) = \mathrm{cone}(\mathcal{T}^*).$$

Then, for every $\mathbf{d} \in T_{\mathcal{X}}(\mathbf{x}^*)$, there exist nonnegative numbers β_i such that

$$\mathbf{d} = \sum_{i\in J}\beta_i\bar{\mathbf{d}}^i, \text{with } \bar{\mathbf{d}}^i \in \mathcal{T}^*. \tag{3.62}$$

Let us recall that, by assumption, $\mathbf{x}^*$ satisfies EMFCQ conditions, and let $\mathbf{d}$ be the direction satisfying (3.3) in point (b) of Definition 3.2.3. Then we have,

$$\begin{aligned} 0 \le\ & \sum_{i\in\bar{\mathcal{J}}}\beta_i\left(\sum_{\ell\in\mathcal{G}^{\mathtt{ext}}} \nu\max\{g_\ell(\mathbf{x}^*),0\}^{\nu-1}\nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p}\nu|h_j(\mathbf{x}^*)|^{\nu-1}\nabla h_j(\mathbf{x}^*)\right)^\top \bar{\mathbf{d}}^i \\ & \left(\sum_{\ell\in\mathcal{G}^{\mathtt{ext}}} \nu\max\{g_\ell(\mathbf{x}^*),0\}^{\nu-1}\nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p}\nu|h_j(\mathbf{x}^*)|^{\nu-1}\nabla h_j(\mathbf{x}^*)\right)^\top \mathbf{d} = \\ & \left(\sum_{\ell\in\mathcal{I}^+(\mathbf{x}^*)\cap\mathcal{G}^{\mathtt{ext}}} \nu\max\{g_\ell(\mathbf{x}^*),0\}^{\nu-1}\nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p}\nu|h_j(\mathbf{x}^*)|^{\nu-1}\nabla h_j(\mathbf{x}^*)\right)^\top \mathbf{d}. \end{aligned}$$

Again by (3.3), $\nabla g_\ell(\mathbf{x}^*)^\top \mathbf{d} < 0$, for all $\ell \in \mathcal{I}^+(\mathbf{x}^*)$, and $\nabla h_j(\mathbf{x}^*)^\top \mathbf{d} = 0$, for all j. Then, we get $\max\{g_\ell(\mathbf{x}^*), 0\} = 0$ for all $\ell \in \mathcal{I}^+(\mathbf{x}^*) \cap \mathcal{G}^{\mathbf{ext}}$, so that $g_\ell(\mathbf{x}^*) \leq 0$ for all $\ell \in \mathcal{G}^{\mathbf{ext}}$. Furthermore, that implies

$$\left(\sum_{\ell \in \mathcal{G}^{\mathbf{ext}}} \nu \max\{g_\ell(\mathbf{x}^*), 0\}^{\nu-1} \nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p} \nu |h_j(\mathbf{x}^*)|^{\nu-1} \nabla h_j(\mathbf{x}^*)\right)^\top \bar{\mathbf{d}} =$$
$$\left(\sum_{j=1}^{p} \nu |h_j(\mathbf{x}^*)|^{\nu-1} \nabla h_j(\mathbf{x}^*)\right)^\top \bar{\mathbf{d}} \geq 0, \text{for all } \bar{\mathbf{d}} \in T_{\mathcal{X}}(\mathbf{x}^*).$$

Using (3.2), we get $h_j(\mathbf{x}^*) = 0$ for all $j = 1, \ldots, p$. Therefore, the point $\mathbf{x}^*$ is feasible.

By simple manipulations, inequality (3.58) can be rewritten as

$$\begin{aligned}\Bigg(&\nabla f(\mathbf{y}_k^i) + \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{y}_k^i) \lambda_\ell(\mathbf{x}_k; \rho_k) \\ &+ \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{y}_k^i)(\lambda_\ell(\mathbf{y}_k^i; \rho_k) - \lambda_\ell(\mathbf{x}_k; \rho_k)) + \sum_{j=1}^{p} \nabla h_j(\mathbf{y}_k^i) \mu_j(\mathbf{x}_k; \rho_k) \\ &+ \sum_{j=1}^{p} \nabla h_j(\mathbf{y}_k^i)(\mu_j(\mathbf{y}_k^i; \rho_k) - \mu_j(\mathbf{x}_k; \rho_k))\Bigg)^\top \bar{\mathbf{d}}^i \geq -\frac{\xi(\hat{\alpha}_k^i)}{\hat{\alpha}_k^i}, \quad \forall i \in \bar{\mathcal{J}}\end{aligned} \tag{3.63}$$

Taking limits for $k \to +\infty$, $k \in \mathcal{K}_\rho^{\mathbf{x},\mathbf{d},\lambda}$ and considering (3.75), (3.81), and (3.82), we get:

$$\left(\nabla f(\mathbf{x}^*) + \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{x}^*) \lambda_\ell^* + \sum_{j=1}^{p} \nabla h_j(\mathbf{x}^*) \mu_j^*\right)^\top \bar{\mathbf{d}}^i \geq 0, \quad \forall i \in \bar{\mathcal{J}}. \tag{3.64}$$

Again, by (3.62) and (3.64), we have, for all $\mathbf{d} \in T_{\mathcal{X}}(\mathbf{x}^*)$,

$$\left(\nabla f(\mathbf{x}^*) + \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{x}^*) \lambda_\ell^* + \sum_{j=1}^{p} \nabla h_j(\mathbf{x}^*) \mu_j^*\right)^\top \mathbf{d} =$$
$$\sum_{i \in J} \beta_i \left(\nabla f(\mathbf{x}^*) + \sum_{\ell=1}^{m} \nabla g_\ell(\mathbf{x}^*) \lambda_\ell^* + \sum_{j=1}^{p} \nabla h_j(\mathbf{x}^*) \mu_j^*\right)^\top \bar{\mathbf{d}}^i \geq 0.$$

Since $\mathbf{x}^*$ is feasible and considering the definition of λ_ℓ^* for $\ell \in \mathcal{I}^+(\mathbf{x}^*)$, $\lambda_\ell^* g_\ell(\mathbf{x}^*) = 0$, for all $\ell = 1, \ldots, m$, and the proof is concluded. □

3.3.4 Implementation Details

In this section, we describe a practical implementation for `LOG-DS`, based on the original implementation of `SID-PSM`.

LOG-DS vs SID-PSM

`LOG-DS` enhances `SID-PSM` with the capability of handling general constraints through a mixed penalty log-barrier approach. Thus, the original structure and algorithmic options of `SID-PSM` implementation are kept. In this subsection, we will provide a general overview of the main features of `SID-PMS`, highlighting the differences with `LOG-DS`. For more details on `SID-PSM`, the original references [35, 36] could be used.

The main difference between `LOG-DS` and `SID-PSM` is the use of a merit function to address constraints, instead of an extreme barrier approach. In `SID-PSM`, only feasible points are evaluated, being the function value set equal to $+\infty$ for infeasible ones. `LOG-DS` allows infeasibility regarding the nonlinear constraints. The merit function also replaces the original objective function through the different algorithmic steps. So, at the Search step, quadratic polynomial models are built for the objective and constraints functions and, after being aggregated into the merit function, are minimized inside a ball with radius directly related to the step-size parameter.

The sets of points used in model computation do not require feasibility regarding the nonlinear constraints, always resulting from previous evaluations of the merit function. No function evaluations are spent solely for the purpose of model building. Depending on the number of points available, minimum Frobenius norm models, quadratic interpolation, or regression approaches can be used [35] to compute the model coefficients.

After the model minimization, `LOG-DS` needs to make a decision on accepting or rejecting the new trial point. Differently from `SID-PSM`, where only simple decrease is required for accepting new points, in `LOG-DS` points are accepted if they satisfy the sufficient decrease condition

$$z^+(\mathbf{x}_{k+1}; \rho_k) \leq z^+(\mathbf{x}_k; \rho_k) - \gamma \alpha_k^2,$$

where $\gamma = 10^{-9}$. The use of a sufficient decrease condition for the acceptance of new points changes the type of globalization strategy used by the algorithm, which is no longer classified as a Generalized Pattern Search method, being now a Generating Set Search (GSS) method.

The algorithm proceeds with an opportunistic Poll Step, accepting the first poll point that satisfies the sufficient decrease condition. Before initiating the polling procedure, previously evaluated points are again used to build a simplex gradient [17], which will be used as an ascent indicator. Poll directions are reordered according to the largest angle made with this ascent indicator, before initiating polling (see [36]). In `LOG-DS`, the simplex gradient is built for the merit function, while in `SID-PSM` the original objective function is considered.

Penalty parameter details

In the initialization of `LOG-DS`, we define the two sets of indices $\mathcal{G}^{\texttt{log}}$ and $\mathcal{G}^{\texttt{ext}}$, considering the values of the inequality constraints at the initial point $\mathbf{x}_0 \in \mathcal{X}$:

$$\mathcal{G}^{\texttt{log}} = \{i \mid g_\ell(\mathbf{x}_0) < 0\} \text{ and } \mathcal{G}^{\texttt{ext}} = \{i \mid g_\ell(\mathbf{x}_0) \geq 0\}.$$

Moreover, we define two penalty parameters, one corresponding to the logarithmic barrier term ($\rho^{\texttt{log}}$) and another one associated with the penalty exterior component ($\rho^{\texttt{ext}}$), and we initialize them as $\rho_0^{\texttt{log}} = 10^{-1}$ and $\rho_0^{\texttt{ext}} = \frac{1}{\max\{|f(x_0)|,10\}}$. Note that we treat ρ^{ext} as $\rho^{\nu-1}$ in equation (3.44). Therefore, we can write the merit function as

$$z^+(\mathbf{x}; \rho_k) = f(\mathbf{x}) - \rho_k^{\texttt{log}} \sum_{\ell \in \mathcal{G}^{\texttt{log}}} \log(-g_\ell(\mathbf{x})) + \frac{1}{\rho_k^{\texttt{ext}}} \left(\sum_{\ell \in \mathcal{G}^{\texttt{ext}}} (\max\{0, g_\ell(\mathbf{x})\})^\nu + \sum_{j=1}^{p} |h_j(\mathbf{x})|^\nu \right). \quad (3.65)$$

The penalty parameters are updated only at unsuccessful iterations, considering two different criteria:

$$\alpha_{k+1} \leq \min\{(\rho_k^{\texttt{log}})^\beta, (g_{\min})_k^2\}, \text{ for updating } \rho_k^{\texttt{log}}; \quad (3.66)$$

$$\alpha_{k+1} \leq \min\{(\rho_k^{\texttt{log}})^\beta, (\rho_k^{\texttt{ext}})^\beta, (g_{\min})_k^2\}, \text{ when updating } \rho_k^{\texttt{ext}}. \quad (3.67)$$

Recall that $(g_{\min})_k$ is the minimum absolute value for the constraints in $\mathcal{G}^{\texttt{log}}$ at iterate x_k. In the implementation, we have considered $\beta = 1 + 10^{-9}$ and $\nu = 2$. Thus, $\rho^{\nu-1} = \rho$ in equation (3.44).

If inequality (3.66) holds, `LOG-DS` uses the following rule

$$\rho_{k+1}^{\texttt{log}} = \zeta \rho_k^{\texttt{log}}, \tag{3.68}$$

whereas if inequality (3.67) holds, `LOG-DS` performs the following update

$$\rho_{k+1}^{\texttt{ext}} = \zeta \rho_k^{\texttt{ext}}, \tag{3.69}$$

in both cases with $\zeta = 10^{-2}$.

The use of two different penalty parameters, for the two different terms of penalization, is a practical need to be able to properly scale the constraints and the different ways they are handled. In particular, in our numerical experience, the logarithmic term seemed not to suffer with the different scales of the objective function, while the exterior penalty seemed to be very sensitive to it. In practice, if the exterior penalty parameter is set too high, the algorithm might be slow at reaching feasible solutions. If it is set too low, the algorithm might not be good at reaching solutions with the best objective function value, even though it might be very capable of attaining feasibility. While scaling the initial exterior penalty parameter with respect to the initial value of the objective function improved the numerical results, it is possible that it might not work for specific problems. Indeed, we are implicitly assuming that the gradient of the objective function is closely related to the objective function value, which might be true for many real problems, but it is certainly not true in general. Scaling the objective function and the constraints for general nonlinear optimization problems is currently an active field of research.

3.3.5 Numerical Experiments

This section is dedicated to the numerical experiments and performance evaluation of the proposed mixed penalty-logarithmic barrier derivative-free optimization algorithm, `LOG-DS`, on a collection of test problems available in the literature.

The numerical experiments exposed in this section use the same collection of test problems and the same performance and data profiles described in Section 3.2.4.

Results and comparison

This subsection aims to demonstrate the good numerical performance of the `LOG-DS` algorithm. It is divided into three parts. First we compare two possible ways of dealing with linear inequalities. They can be either addressed explicitly by using strategies to conform the directions of the Poll Step to the tangent cone of ε-active constraints, or they can be treated within the penalty-barrier function as the other nonlinear inequalities. After selecting the most performative strategy among the last two options, we make sure that the MPB is more efficient than the original extreme barrier used in `SID-PSM`. In order to do so, we select a subset of problems with a strictly feasible point with respect to the inequality constraints and we test the algorithm on such collections using the two approaches. Finally, we explore the efficiency of the proposed method against state-of-the-art solvers, i.e. `X-LOG-DFL` and `NOMAD`.

Comparison between strategies for linear constraints

In this subsection, our focus lies on evaluating the performance of `LOG-DS` using two distinct approaches for managing linear inequality constraints, other than bounds. The first approach addresses each linear inequality constraint as a general nonlinear inequality, i.e. through the penalty approach discussed previously. The second approach entails addressing the linear inequality constraints via an extreme barrier method, adjusting the directions in accordance with the geometry of the feasible region.

The works by Lucidi et al. [76] and Lewis and Torczon [66] propose methods for computing directions conforming to linear inequality constraints but do not consider degeneracy. Abramson et al. [1] provide a detailed algorithm for generating the set of desired directions, regardless of whether the constraints are degenerate or not.

We normalize the linear constraints, i.e., given the rows $\mathbf{a}_i$ of the matrix $\mathbf{A} \in \mathbb{R}^{p\times n}$ and the relative constants b_i, we perform the following transformation

$$\overline{\mathbf{a}}_i = \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|}, \quad \bar{b}_i = \frac{b_i}{\|\mathbf{a}_i\|}, \quad i = 1, \dots, p. \tag{3.70}$$

The ε-active index set is computed using the matrix $\bar{A} = [\mathbf{a_1} \ \dots \ \mathbf{a_p}]$ and the vector $\mathbf{b} = [b_1 \ \dots b_p]$..

To compute the set of directions $\mathcal{P}_k$ that conform to the geometry of the nearby constraints, we use the algorithm proposed in [1, Alg. 4.4.2]. The latter is divided into two parts: the first constructs the index set corresponding to ε-active non-redundant constraints, and the second the set of directions $\mathcal{P}_k$, which include the generators of the cone $T_{\mathcal{X}}(\mathbf{x}_k, \varepsilon)$.

It is important to understand why we are comparing the two strategies. First, linear constraints might not be explicitly given for a black-box type problem, making it a hard task to conform the directions to the linear constraints. In such cases, we would be forced to treat the linear inequalities with a mixed penalty. Furthermore, the logarithmic barrier approach is well-known for handling linear constraints very efficiently, especially in the presence of a large number of constraints. Finally, conforming the directions to the nearby linear constraints might affect the geometry of the generated points, impacting the quality of the surrogate models built to improve the performance of the algorithm. Using the penalty approach allows us to keep using suitable sets of directions, which can be generated to produce trial points with a good geometry for building surrogate models.

We tested different values of ε to identify active constraints, obtaining the best results with $\varepsilon = 10^{-5}$. We used $\mathcal{P}_k = [\mathbf{1} \ \ -\mathbf{1} \quad \mathbb{I}_n \ \ -\mathbb{I}_n]$ as the default set of directions, for each k. Then we compared the performance of the algorithm using the two different strategies to deal with linear constraints. Figure 3.9 depicts such comparison, considering a maximum number of 2000 function evaluations and a minimum step-size tolerance equal to 10^{-8}.

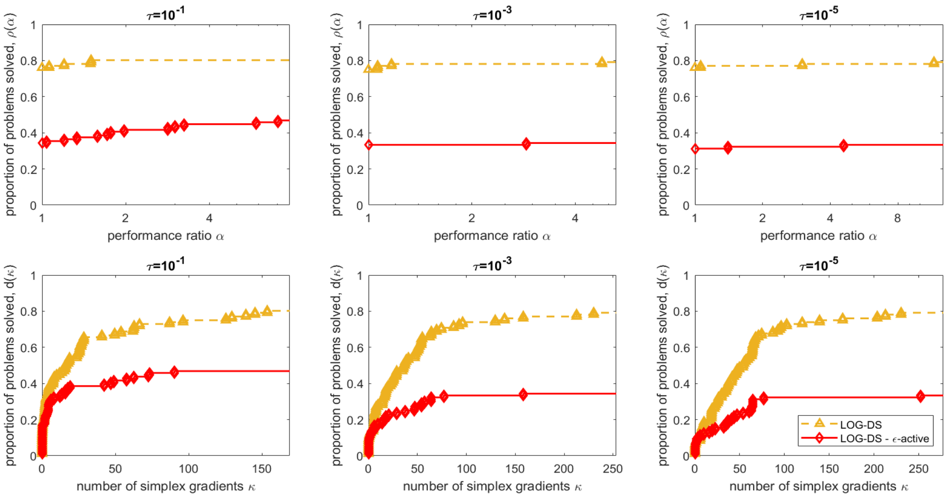

Figure 3.9. Performance (on top) and data (on bottom) profiles comparing `LOG-DS` using two different approaches to address linear inequality constraints.

As we can see, the performance of the `LOG-DS` algorithm when addressing the linear inequality constraints within the penalty approach outperforms the competing strategy, addressing the linear constraints directly. Therefore, in the rest of the work, the experiments will be carried out using the winning strategy. Note that the significant difference in the performance might be due to the specific choice of the tested problems and/or the specific strategy used to conform the directions to the linear constraints, rather than to a flaw in the approach itself.

Comparison with the original Extreme Barrier

We are proposing an alternative strategy to address constraints within the `SID-PSM` algorithm. Thus, we start by illustrating that the use of a mixed penalty-logarithmic barrier is competitive against the extreme barrier approach. The latter can only be adopted for problems without equality constraints and for which a strictly feasible point is given as initialization, so we selected a subset of problems satisfying these conditions. The subset consists of a total of 28 problems, highlighted in Table 3.1.

Figure 3.10 presents the comparison between `LOG-DS`, which exploits the MPB, and the original `SID-PSM`, which employs an extreme barrier approach. The default values of `SID-PSM` were considered for both algorithms, allowing a maximum number of 2000 function evaluations and a minimum step-size tolerance equal to 10^{-8}.

As Figure 3.10 shows, `LOG-DS` presents a better performance than `SID-PSM`, especially when a higher precision is considered. Furthermore, the possibility of initializing `LOG-DS` with infeasible points allows to handle a wider class of practical problems.

Comparison with state-of-the-art solvers

This subsection focuses on comparing `LOG-DS` against state-of-the-art DFO solvers that are able to address general nonlinear constraints. Comparisons were made with MADS [8], implemented in the well-known `NOMAD` package (version 4), which can be freely obtained at `https://www.gerad.ca/en/software/nomad` [9]. Additionally, the `X-LOG-DFL` algorithm [21], available through the DFL

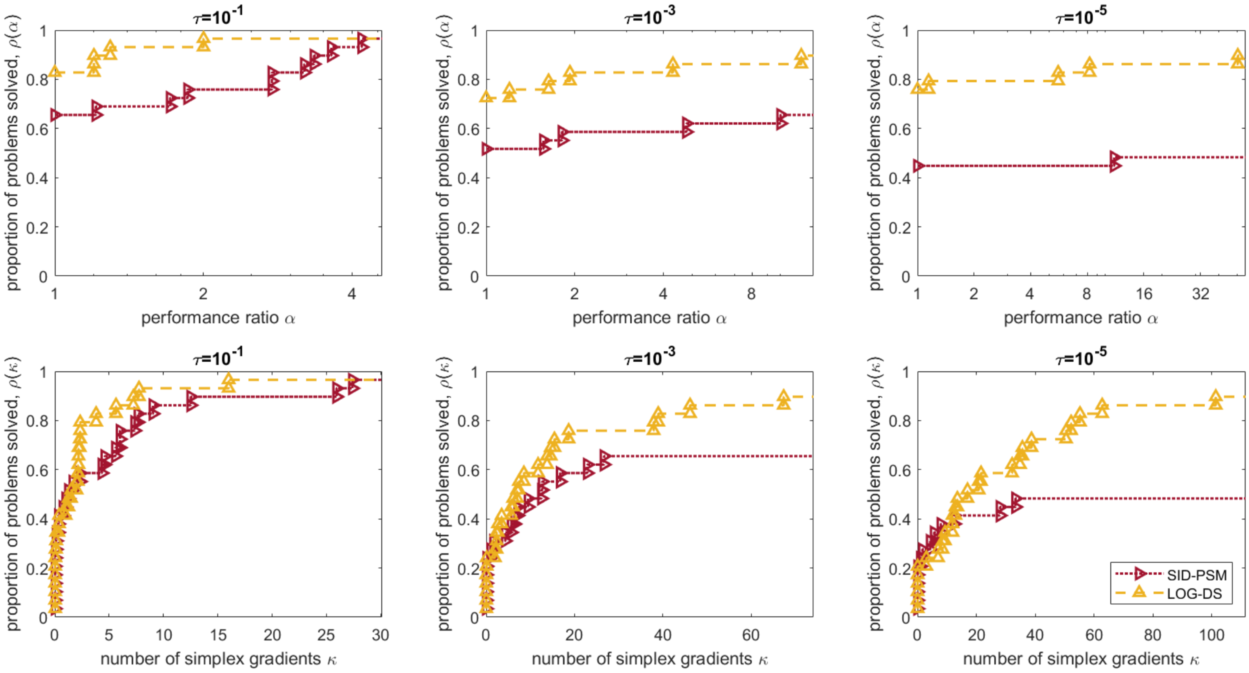

Figure 3.10. Performance (on top) and data (on bottom) profiles comparing `LOG-DS` and `SID-PSM`.

library as the `LOGDFL` package at `https://github.com/DerivativeFreeLibrary/LOGDFL`, was also tested. Comparison with `LOG-DFL` [21] is particularly relevant since it uses the same merit function `LOG-DS` does. Default settings were considered for all codes and results, reported in Figure 3.11, were obtained for a budget of 2000 function evaluations.

It can be observed that `LOG-DS` presents the best performance, for any of the three precision levels considered, both in terms of efficiency and robustness, across the different computational budgets.

Figure 3.12 compares the different solvers considering the subset of problems with only inequality constraints (61 out of 96 problems), again allowing a maximum of 2000 function evaluations. Once more, `LOG-DS` is clearly the solver with the best performance.

In summary, considering the outcomes of the different numerical experiments, we can conclude that `LOG-DS` is the most efficient and robust solver across different scenarios, making it the top-performing choice.

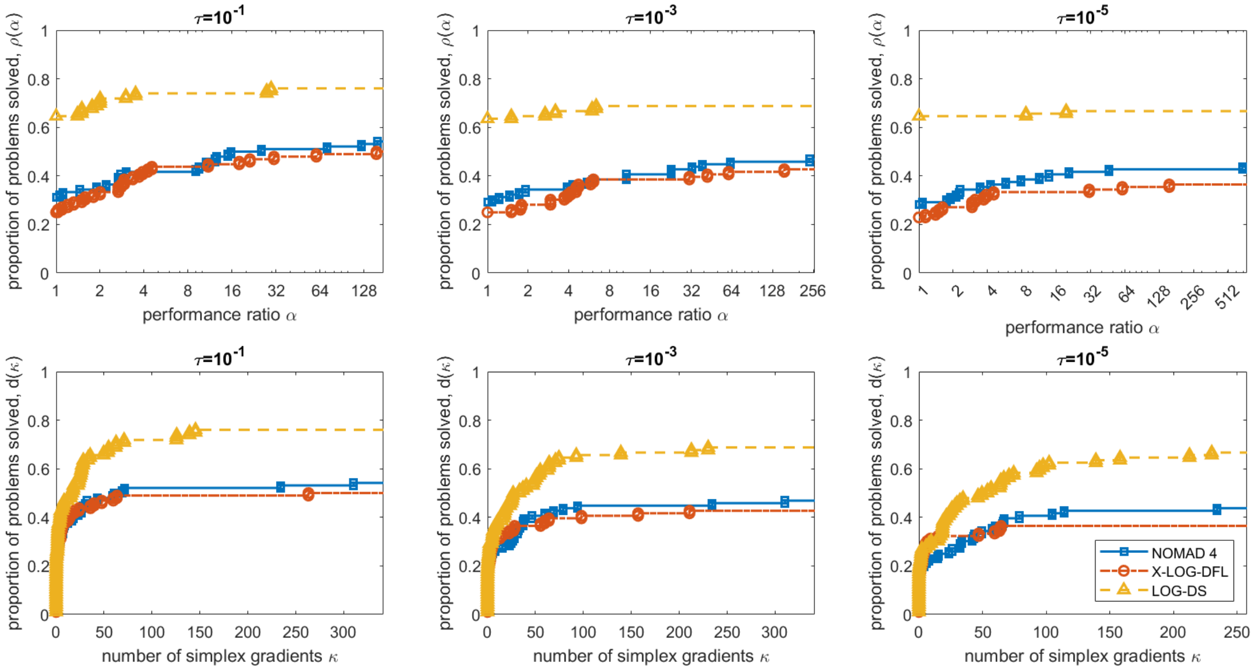

Figure 3.11. Performance (on top) and data (on bottom) comparing `LOG-DS`, `NOMAD`, and `X-LOG-DFL`, on the complete problem collection.

3.4 Technical results: boundedness of multipliers

In the analysis of both `LOG-DFL` and `LOG-DS`, specifically in the proof of the main convergence results, i.e. Theorem 3.2.9 and Theorem 3.3.10 respectively, we delayed proving the boundedness of the sequences of multipliers. The techniques used to prove such property for the two algorithms are quite similar to each other, thus, the results are presented together in this section, and only the proofs related to `LOG-DS` are provided. The proof of Theorem 3.4.7 can be found in [21].

Before diving into the main result of this section, that is, Theorem 3.4.6, we have to prove some intermediate results. First we provide the following two Lemmas to justify a technical step in Proposition 3.4.4.

Lemma 3.4.1. *Let* $a \in \mathbb{R}_+$, $b \in \mathbb{R}_+$, $a + b > 0$ *and* $p \in [0, 1]$. *It results*

$$(a+b)^p \le a^p + b^p.$$

Proof. We have $\dfrac{a}{a+b} \le 1$ and $\dfrac{b}{a+b} \le 1$, so

$$\begin{aligned}(a+b)^p &= \frac{(a+b)}{(a+b)^{1-p}} = \frac{a}{(a+b)^{1-p}} + \frac{b}{(a+b)^{1-p}} \\ &= a^p \left(\frac{a}{a+b}\right)^{1-p} + b^p \left(\frac{b}{a+b}\right)^{1-p} \\ &\le a^p + b^p.\end{aligned}$$

□

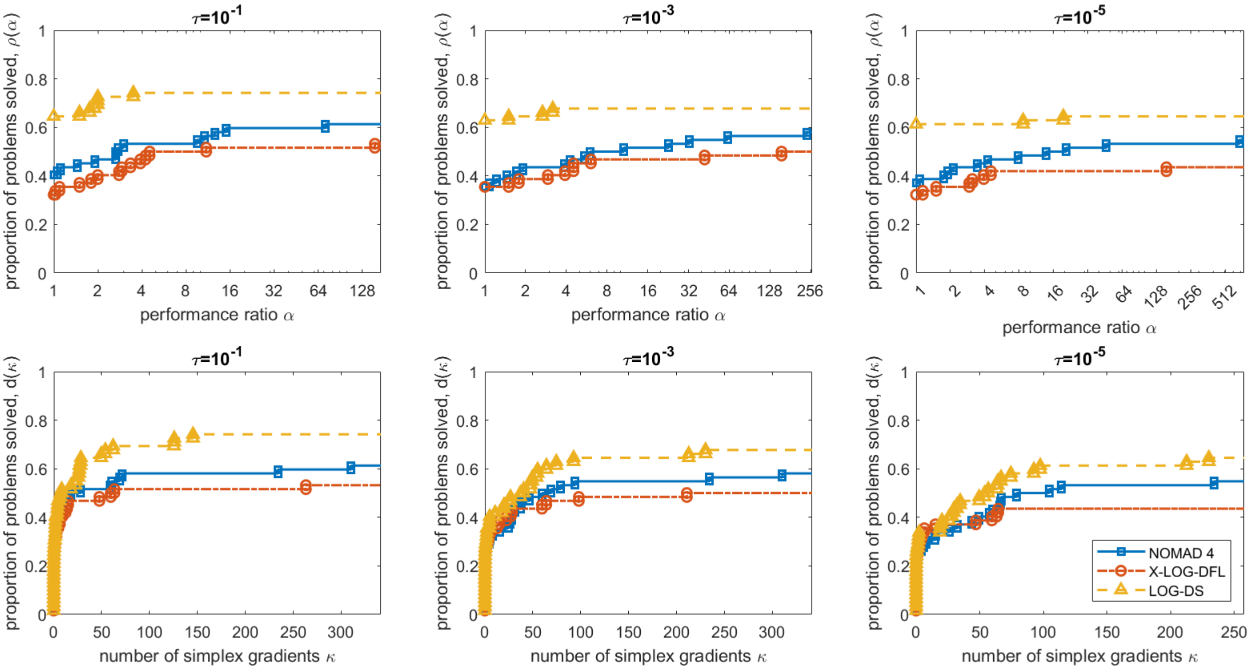

Figure 3.12. Performance (on top) and data (on bottom) profiles comparing `LOG-DS`, `NOMAD`, and `LOG-DFL`, on the subset of problems with only inequality constraints.

Lemma 3.4.2. *Let* $a \in \mathbb{R}$*,* $b \in \mathbb{R}$ *and* $p \in [0,1]$*. The following inequalities hold*

$$||a|^p - |b|^p| \leq |a-b|^p, \tag{3.71}$$

and

$$|\max\{a,0\}^p - \max\{b,0\}^p| \leq |a-b|^p. \tag{3.72}$$

Proof. To prove (3.71) consider

$$\begin{array}{rcl} |a|^p & = & |a-b+b|^p \\ \text{(Triangular inequality)} & \leq & (|a-b|+|b|)^p \\ \text{(Lemma 3.4.1)} & \leq & |a-b|^p + |b|^p \end{array} \tag{3.73}$$

On the other hand,

$$\begin{array}{rcl} |b|^p & = & |b-a+a|^p \\ \text{(Triangular inequality)} & \leq & (|b-a|+|a|)^p \\ \text{(Lemma 3.4.1)} & \leq & |a-b|^p + |a|^p \end{array} \tag{3.74}$$

Then, from inequalities (3.73) and (3.74) we derive inequality (3.71).
Now, to prove (3.72), we will analyze four different cases:

i) If $a \leq 0$ and $b \leq 0$, the result holds trivially.

ii) If $a \leq 0$ and $b > 0$, then $|-b^p| = b^p \leq |b-a|^p = |a-b|^p$.

iii) If $a > 0$ and $b \leq 0$, then $|a|^p = a^p \leq |a-b|^p$.

iv) If $a > 0$ and $b > 0$, using (3.71) we conclude that:

$$|\max\{a,0\}^p - \max\{b,0\}^p| = |a^p - b^p| = ||a|^p - |b|^p| \leq |a-b|^p.$$

□

The following two propositions allow to bound the estimations of the multipliers in $\mathbf{y}_k^i$ with the ones in $\mathbf{x}_k$ for all $i \in \mathcal{J}_k$. Note that such bounds are necessary since, if we let $k \to \infty$ in equation (3.58), we might get different limits on the sequences $\{\lambda_\ell(\mathbf{y}_k^i;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathtt{x}}}$ for different i, so that the arguments used in Theorem 3.4.6 would no longer be valid.

Proposition 3.4.3. *Consider the conditions of Theorem 3.3.10, let $\mathcal{K}_\rho^{\mathtt{x}}$ be the sequence considered in the proof of the theorem, $\mathbf{y}_k^i$ defined as in (3.56), and let*

$$\lambda_\ell(\mathbf{x};\rho) = \frac{\rho}{-g_\ell(\mathbf{x})}, \quad \textit{for all } \ell \in \mathcal{G}^{\log}.$$

Then we have

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathtt{x}}}} |\lambda_\ell(\mathbf{x}_k;\rho_k) - \lambda_\ell(\mathbf{y}_k^i;\rho_k)| = 0, \quad \ell \in \mathcal{G}^{\log}, \quad \forall\, i \in \mathcal{J}_k \tag{3.75}$$

Proof. For all $\ell \in \mathcal{G}^{\log}$, considering the expression of $\lambda_\ell(\cdot)$

$$\begin{aligned}
\left|\frac{\rho_k}{-g_\ell(\mathbf{x}_k)} - \frac{\rho_k}{-g_\ell(\mathbf{y}_k^i)}\right| &= \rho_k\left|\frac{g_\ell(\mathbf{x}_k) - g_\ell(\mathbf{y}_k^i)}{(-g_\ell(\mathbf{y}_k^i))(-g_\ell(\mathbf{x}_k))}\right| \\
&= \rho_k \frac{\left|\nabla g_\ell\left(\mathbf{u}_k^i\right)^\top\left(\mathbf{x}_k - \mathbf{y}_k^i\right)\right|}{\left|g_\ell\left(\mathbf{y}_k^i\right)\right|\left|g_\ell\left(\mathbf{x}_k\right)\right|} \\
&\leq \rho_k \frac{\left\|\nabla g_\ell\left(\mathbf{u}_k^i\right)\right\|\left\|\mathbf{y}_k^i - \mathbf{x}_k\right\|}{\left|g_\ell\left(\mathbf{y}_k^i\right)\right|\left|g_\ell\left(\mathbf{x}_k\right)\right|},
\end{aligned} \tag{3.76}$$

where $\mathbf{u}_k^i = \mathbf{x}_k + \tilde{t}_k^i(\mathbf{y}_k^i - \mathbf{x}_k)$ with $\tilde{t}_k^i \in (0,1)$.

Then, $c_1 > 0$ exists such that

$$\begin{aligned}
\rho_k\left\|\nabla g_\ell(\mathbf{u}_k^i)\right\|\frac{\|\mathbf{y}_k^i - \mathbf{x}_k\|}{\left|g_\ell\left(\mathbf{y}_k^i\right)\right|\left|g_\ell\left(\mathbf{x}_k\right)\right|} &\leq \rho_k c_1 \frac{\|\mathbf{y}_k^i - \mathbf{x}_k\|}{|g_\ell(\mathbf{y}_k^i)||g_\ell(\mathbf{x}_k)|} \\
&= \rho_k c_1 \frac{\|\mathbf{x}_k + t_k^i\hat{\alpha}_k^i\mathbf{d}_k^i - \mathbf{x}_k\|}{|g_\ell(\mathbf{y}_k^i)||g_\ell(\mathbf{x}_k)|} \\
&= \rho_k c_1 \frac{\|t_k^i\hat{\alpha}_k^i\mathbf{d}_k^i\|}{|g_\ell(\mathbf{y}_k^i)||g_\ell(\mathbf{x}_k)|} \\
&= \rho_k c_1 \frac{t_k^i\hat{\alpha}_k^i\|\mathbf{d}_k^i\|}{|g_\ell(\mathbf{y}_k^i)||g_\ell(\mathbf{x}_k)|}.
\end{aligned} \tag{3.77}$$

Now, we will prove that there is another constant $c_2 > 0$ such that

$$\frac{1}{|g_\ell(\mathbf{y}_k^i)|} \leq c_2\frac{1}{|g_\ell(\mathbf{x}_k)|}. \tag{3.78}$$

Suppose, in order to arrive to a contradiction, that c_2 does not exist. This would imply that there exists $\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}} \subseteq \mathcal{K}_\rho^{\mathtt{x}}$ such that

$$\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{\frac{1}{|g_\ell(\mathbf{y}_k^i)|}}{\frac{1}{|g_\ell(\mathbf{x}_k)|}} = \lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{|g_\ell(\mathbf{x}_k)|}{|g_\ell(\mathbf{y}_k^i)|} = +\infty. \tag{3.79}$$

Let us consider the case where

$$\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} |g_\ell(\mathbf{x}_k)| = 0.$$

Since $g_\ell(\mathbf{x}_k) < 0$ and $g_\ell(\mathbf{y}_k^i) < 0$ for all $k \in \mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}$, by (3.79) there exists $\overline{k} \in \mathbb{N}$ such that, for all $k \geq \overline{k}$, $k \in \mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}$, we have

$$-g_\ell(\mathbf{x}_k) > -g_\ell(\mathbf{y}_k^i) = -g_\ell(\mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i).$$

Using the Lipschitz continuity of g_ℓ, $\ell = 1, \ldots, m$ and the fact that $\|\mathbf{d}_k^i\| = 1$ for all $i \in \mathcal{J}_k$, we get

$$-g_\ell(\mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i) \geq -g_\ell(\mathbf{x}_k) - L_{g_\ell}\|t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i\| = -g_\ell(\mathbf{x}_k) - L_{g_\ell} t_k^i \hat{\alpha}_k^i.$$

The definition of $\mathcal{K}_\rho$ guarantees that

$$\alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\mathtt{min}})_k^2\}, \qquad \alpha_{k+1} = \theta_\alpha \alpha_k,$$

so that, for all $k \in \mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}$

$$\alpha_k \leq \frac{\min\{\rho_k^\beta, (g_{\mathtt{min}})_k^2\}}{\theta_\alpha} \tag{3.80}$$

Hence, since $\hat{\alpha}_k^i \leq \alpha_k$, we have

$$-g_\ell(\mathbf{x}_k) - L_{g_\ell} t_k^i \hat{\alpha}_k^i \geq -g_\ell(\mathbf{x}_k) - L_{g_\ell} t_k^i \frac{1}{\theta_\alpha}(g_\ell(\mathbf{x}_k))^2, \quad \forall k \geq \overline{k},\ k \in \mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}$$

Thus,

$$\begin{aligned}
\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{-g_\ell(\mathbf{x}_k)}{-g_\ell(\mathbf{y}_k^i)} &= \lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{-g_\ell(\mathbf{x}_k)}{-g_\ell(\mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i)} \\
&\leq \lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{-g_\ell(\mathbf{x}_k)}{-g_\ell(\mathbf{x}_k) - L_{g_\ell} t_k^i \frac{1}{\theta_\alpha}(g_\ell(\mathbf{x}_k))^2} = 1,
\end{aligned}$$

which leads to a contradiction, proving (3.78).

Now, by considering the other case

$$\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} |g_\ell(\mathbf{x}_k)| = c < +\infty,$$

we have

$$\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{-g_\ell(\mathbf{x}_k)}{-g_\ell(\mathbf{y}_k^i)} = \lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathtt{x},\mathtt{g}}}} \frac{-g_\ell(\mathbf{x}_k)}{-g_\ell(\mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i)} < +\infty.$$

Again, this leads to a contradiction, proving (3.78).

Hence, the existence of the constant $c_2 > 0$, (3.78), and recalling that $\hat{\alpha}_k^i \leq \alpha_k$, allow us to write

$$\frac{\rho_k c_1 t_k^i \hat{\alpha}_k^i}{|g_\ell(\mathbf{y}_k^i)||g_\ell(\mathbf{x}_k)|} \quad \leq \quad \frac{\rho_k c_1 c_2 t_k^i \alpha_k}{|g_\ell(\mathbf{x}_k)|^2}.$$

The instructions of Step 3 imply that $\mathbf{x}_{k+1} = \mathbf{x}_k$, so that $(g_{\min})_k = \min_{\ell \in \mathcal{G}^{\texttt{log}}} \{|g_\ell(\mathbf{x}_{k+1})|\} = \min_{\ell \in \mathcal{G}^{\texttt{log}}} \{|g_\ell(\mathbf{x}_k)|\}$. Recalling (3.80), we get

$$\frac{\rho_k \alpha_k}{(g_{\texttt{min}})_k^2} \leq \frac{\rho_k}{\theta_\alpha}.$$

Then, recalling Theorem 3.3.8, (3.75) is proved. □

Proposition 3.4.4. *Consider the conditions of Theorem 3.3.10, let* $\mathcal{K}_\rho^{\mathrm{x}}$ *be the sequence considered in the proof of the theorem,* $\mathbf{y}_k^i$ *defined as in* (3.56)*, and let*

$$\lambda_\ell(\mathbf{x};\rho) = \nu \left(\frac{\max\{g_\ell(\mathbf{x}), 0\}}{\rho} \right)^{\nu-1}, \quad \textit{for all } \ell \in \mathcal{G}^{\texttt{ext}},$$

$$\mu_\ell(\mathbf{x};\rho) = \nu \left(\frac{|h_j(\mathbf{x})|}{\rho} \right)^{\nu-1}, \quad j = 1, \ldots, p.$$

Then we have

$$\lim_{\substack{k \to +\infty \\ k \in \mathcal{K}_\rho^{\mathrm{x}}}} |\lambda_\ell(\mathbf{x}_k;\rho_k) - \lambda_\ell(\mathbf{y}_k^i;\rho_k)| = 0, \quad \ell \in \mathcal{G}^{\texttt{ext}}, \quad \forall\, i \in \mathcal{J}_k, \tag{3.81}$$

and

$$\lim_{\substack{k \to +\infty \\ k \in \mathcal{K}_\rho^{\mathrm{x}}}} |\mu_j(\mathbf{x}_k;\rho_k) - \mu_j(\mathbf{y}_k^i;\rho_k)| = 0, \quad j = 1, \ldots, p, \quad \forall i \in \mathcal{J}_k. \tag{3.82}$$

Proof. We first prove (3.81). For all $\ell \in \mathcal{G}^{\texttt{ext}}$, considering the expression of $\lambda_\ell(\cdot)$

$$\begin{array}{rcl}
\left| \dfrac{\nu}{\rho_k^{\nu-1}} \left(\max\{g_\ell(\mathbf{x}_k), 0\}\right)^{\nu-1} \right. & - & \left. \dfrac{\nu}{\rho_k^{\nu-1}} \left(\max\{g_\ell(\mathbf{y}_k^i), 0\}\right)^{\nu-1} \right| = \\
 & = & \dfrac{\nu}{\rho_k^{\nu-1}} \left| \max\{g_\ell(\mathbf{x}_k), 0\}^{\nu-1} - \max\{g_\ell(\mathbf{x}_k) + \nabla g_\ell(\mathbf{u}_k^i)^\top (\mathbf{x}_k - \mathbf{y}_k^i), 0\}^{\nu-1} \right| \\
\text{(Lemma 3.4.2 – (3.72))} & \leq & \dfrac{\nu}{\rho_k^{\nu-1}} \left| g_\ell(\mathbf{x}_k) - g_\ell(\mathbf{x}_k) - \nabla g_\ell(\mathbf{u}_k^i)^\top (\mathbf{x}_k - \mathbf{y}_k^i) \right|^{\nu-1} \\
 & = & \dfrac{\nu}{\rho_k^{\nu-1}} |\nabla g_\ell(\mathbf{u}_k^i)^\top (\mathbf{x}_k - \mathbf{y}_k^i)|^{\nu-1} \leq \dfrac{\nu}{\rho_k^{\nu-1}} \left\| \nabla g_\ell(\mathbf{u}_k^i) \right\|^{\nu-1} \left\| \mathbf{x}_k - \mathbf{y}_k^i \right\|^{\nu-1} \\
 & \leq & c_3 \dfrac{\nu}{\rho_k^{\nu-1}} \left\| \left(\mathbf{x}_k - (\mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i)\right) \right\|^{\nu-1} = c_3 \dfrac{\nu}{\rho_k^{\nu-1}} (t_k^i \hat{\alpha}_k^i)^{\nu-1} \left\| \mathbf{d}_k^i \right\|^{\nu-1} \\
 & \leq & c_3 \dfrac{\nu}{\rho_k^{\nu-1}} (\hat{\alpha}_k^i)^{\nu-1} \leq c_3 \nu \left(\dfrac{\alpha_k}{\rho_k} \right)^{\nu-1} \leq c_3 \nu \left(\dfrac{\rho_k^{\beta-1}}{\theta_\alpha} \right)^{\nu-1} \\
 & = & c_3 \nu \theta_\alpha^{1-\nu} \rho_k^{(\beta-1)(\nu-1)},
\end{array}$$

where $\mathbf{u}_k^i = \mathbf{x}_k + \tilde{t}_k^i (\mathbf{y}_k^i - \mathbf{x}_k)$ with $\tilde{t}_k^i \in (0,1)$, and $c_3 > 0$. Thus, using $\beta > 1$, $\nu \in (1, 2]$, $\|\mathbf{d}_k^i\| = 1$ for all $i \in \mathcal{J}_k$, and recalling Theorem 3.3.8, (3.81) is proved.

We now prove (3.82). For all $j = 1, \dots, q$, considering the expression of $\mu_j(\cdot)$ and recalling $\nu \in (1, 2]$

$$\begin{aligned}
\left| \nu \left| \frac{h_j(\mathbf{x}_k)}{\rho_k} \right|^{\nu-1} - \nu \left| \frac{h_j(\mathbf{y}_k^i)}{\rho_k} \right|^{\nu-1} \right| &= \frac{\nu}{\rho_k^{\nu-1}} \left| |h_j(\mathbf{x}_k)|^{\nu-1} - \left| h_j(\mathbf{x}_k) + \nabla h_j\left(\mathbf{u}_k^j\right)^\top (\mathbf{y}_k^i - \mathbf{x}_k) \right|^{\nu-1} \right| \\
\text{(Lemma 3.4.2 – (3.71))} \quad &\leq \frac{\nu}{\rho_k^{\nu-1}} \left| h_j(\mathbf{x}_k) - h_j(\mathbf{x}_k) - \nabla h_j\left(\mathbf{u}_k^j\right)^\top (\mathbf{y}_k^i - \mathbf{x}_k) \right|^{\nu-1} \\
&= \frac{\nu}{\rho_k^{\nu-1}} \left| \nabla h_j(\mathbf{u}_k^j)^\top (\mathbf{y}_k^i - \mathbf{x}_k) \right|^{\nu-1} \\
&\leq \frac{\nu}{\rho_k^{\nu-1}} \left\| \nabla h_j(\mathbf{u}_k^j) \right\|^{\nu-1} \left\| (\mathbf{y}_k^i - \mathbf{x}_k) \right\|^{\nu-1}, \qquad (3.83)
\end{aligned}$$

where $\mathbf{u}_k^j = \mathbf{x}_k + \tilde{t}_k^i(\mathbf{y}_k^i - \mathbf{x}_k)$, with $\tilde{t}_k^i \in (0, 1)$ Now, recalling that $h_j(\cdot)$, $j = 1, \dots, p$ are continuously differentiable functions and $\mathbf{y}_k^i = \mathbf{x}_k + t_k^i \hat{\alpha}_k^i \mathbf{d}_k^i$, with $t_k^i \in (0, 1)$ and $\|\mathbf{d}_k^i\| = 1$, from (3.83) and the fact that $\hat{\alpha}_k^i \leq \alpha_k$ we can write

$$\begin{aligned}
\left| \nu \left| \frac{h_j(\mathbf{x}_k)}{\rho_k} \right|^{\nu-1} - \nu \left| \frac{h_j(\mathbf{y}_k^i)}{\rho_k} \right|^{\nu-1} \right| &\leq \frac{\nu}{\rho_k^{\nu-1}} c_4 \left(t_k^i \hat{\alpha}_k^i \right)^{\nu-1} \\
&\leq c_4 \frac{\nu}{\rho_k^{\nu-1}} \left(t_k^i \alpha_k \right)^{\nu-1} \leq c_4 \nu \left(\frac{\alpha_k}{\rho_k} \right)^{\nu-1} \\
&\leq c_4 \nu \left(\frac{\rho_k^{\beta-1}}{\theta_\alpha} \right)^{\nu-1} = c_4 \nu \theta_\alpha^{1-\nu} \rho_k^{(\beta-1)(\nu-1)}.
\end{aligned}$$

Given that $\beta > 1$, $\nu \in (1, 2]$, and recalling Theorem 3.3.8, we can conclude that (3.82) holds.

□

The following Lemma is the piece completing the picture of requirements to delve into the proof of Theorem 3.4.6. The result is technical and relatively intuitive, and its proof, which can be found in [19], does not provide further insights on the method we have proposed, so it is omitted in this thesis.

Lemma 3.4.5. *Let $\{a_k^i\}_{k \in \mathbb{N}}$, $i = 1, \dots, \ell$, be sequences of scalars. Two cases can occur:*

(i) It results $\lim_{k \to +\infty} a_k^i = 0$, $i = 1, \dots, \ell$. In particular, all the sequences are bounded;

(ii) An index $j \in \{1, \dots, \ell\}$, an infinite index set $K_j \subseteq \{0, 1, \dots\}$ and a positive scalar $\bar{a}_j$ exist such that

$$|a_k^j| > \bar{a}_j > 0, \quad \forall k \in K_j,$$

i.e., at least one sequence is not convergent to zero. Then, there exists an index $s \in \{1, \dots, \ell\}$ and an infinite subset $K \subseteq \mathbb{N}$ such that:

$$\lim_{\substack{k \to +\infty, \\ k \in K}} \frac{a_k^i}{|a_k^s|} = z_i, \quad |z_i| < +\infty, \quad i = 1, \dots, \ell, \qquad (3.84)$$

i.e., all the sequences $\left\{ \frac{a_k^i}{|a_k^s|} \right\}_{k \in K}$ are bounded.

We are now ready to present the following theorem, proving the boundedness of the sequences of multipliers.

Theorem 3.4.6. *Consider the conditions of Theorem 3.3.10, let $\mathcal{K}_\rho^{\mathrm{x}}$ be the sequence considered in the proof of the theorem. Then, the following sequences are bounded*

$$\begin{aligned}
&\{\lambda_\ell(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathrm{x}}}, \quad \ell=1,\dots,m,\\
&\{\mu_j(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathrm{x}}}, \quad j=1,\dots,p.
\end{aligned}$$

Proof. Recalling the expression of $\lambda_\ell(\cdot)$ for all $\ell=1,\dots,m$, and the expression of $\mu_j(\cdot)$ for all $j=1,\dots,p$, and by simple manipulations, we can rewrite inequality (3.58) as

$$\begin{aligned}
&\Bigg(\nabla f(\mathbf{y}_k^i)+\sum_{\ell=1}^m \nabla g_\ell(\mathbf{y}_k^i)\lambda_\ell(\mathbf{x}_k;\rho_k)+\\
&+\sum_{\ell=1}^m \nabla g_\ell(\mathbf{y}_k^i)(\lambda_\ell(\mathbf{y}_k^i;\rho_k,)-\lambda_\ell(\mathbf{x}_k;\rho_k))+\sum_{j=1}^p \nabla h_j(\mathbf{y}_k^i)\mu_j(\mathbf{x}_k;\rho_k)+\\
&+\sum_{j=1}^p \nabla h_j(\mathbf{y}_k^i)(\mu_j(\mathbf{y}_k^i;\rho_k)-\mu_j(\mathbf{x}_k;\rho_k))\Bigg)^\top \mathbf{d}_k \geq -\frac{\xi(\hat{\alpha}_k^i)}{\hat{\alpha}_k^i}, \quad \forall i\in\mathcal{J}_k \text{ and } k\in\mathcal{K}_\rho^{\mathrm{x}},
\end{aligned}\tag{3.85}$$

where $\mathbf{y}_k^i=\mathbf{x}_k+t_k^i\hat{\alpha}_{\mathbf{d}_k}\mathbf{d}_k^i$, with $t_k^i\in(0,1)$ and $\hat{\alpha}_k^i\leq\alpha_k$. Let

$$\begin{aligned}
\{a_k^1,\dots,a_k^m\}&=\{\lambda_1(\mathbf{x}_k;\rho_k),\dots,\lambda_m(\mathbf{x}_k;\rho_k)\},\\
\{a_k^{m+1},\dots,a_k^{m+p}\}&=\{\mu_1(\mathbf{x}_k;\rho_k),\dots,\mu_p(\mathbf{x}_k;\rho_k)\}.
\end{aligned}$$

Assume, by contradiction, that there exists at least one index $l\in\{1,\dots,m+p\}$ such that

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathrm{x}}}} |a_k^l|=+\infty. \tag{3.86}$$

Hence, the sequence $\{a_k^i\}$, $i=1,\dots,m+p$, cannot be all convergent to zero. Then, from Lemma 3.4.5, there exists an infinite subset $\mathcal{K}_\rho^{\mathrm{x,a}}\subseteq\mathcal{K}_\rho^{\mathrm{x}}$ and an index $s\in\{1,\dots,m+p\}$ such that,

$$\lim_{\substack{k\to+\infty\\ k\in\mathcal{K}_\rho^{\mathrm{x,a}}}} \frac{a_k^i}{|a_k^s|}=z_i, \quad |z_i|<+\infty, \quad i=1,\dots,m+p \tag{3.87}$$

If the index l satisfying (3.86) is unique, then $s=l$. If multiple indices satisfy the equation, then s is selected as one of the indices such that $\{a_k^s\}_{k\in\mathcal{K}_\rho^{\mathrm{x,a}}}$ tends to $+\infty$ faster than the others. Note also that

$$z_s=1, \qquad \text{and} \qquad |a_k^s|\to+\infty. \tag{3.88}$$

Dividing the relation (3.85) by $|a_k^s|$, we have

$$\begin{aligned}
&\Bigg(\frac{\nabla f(\mathbf{y}_k^i)}{|a_k^s|}+\sum_{\ell=1}^m \frac{\nabla g_\ell(\mathbf{y}_k^i)a_k^\ell}{|a_k^s|}\\
&+\sum_{\ell=1}^m \nabla g_\ell(\mathbf{y}_k^i)\frac{\lambda_\ell(\mathbf{y}_k^i;\rho_k)-\lambda_\ell(\mathbf{x}_k;\rho_k)}{|a_k^s|}+\sum_{j=1}^p \frac{\nabla h_j(\mathbf{y}_k^i)a_k^{m+j}}{|a_k^s|}\\
&+\sum_{j=1}^p \nabla h_j(\mathbf{y}_k^i)\frac{\mu_j(\mathbf{y}_k^i;\rho_k)-\mu_j(\mathbf{x}_k;\rho_k)}{|a_k^s|}\Bigg)^\top \mathbf{d}_k^i \geq -\frac{\xi(\hat{\alpha}_k^i)}{\hat{\alpha}_k^i|a_k^s|}, \quad \forall i\in\mathcal{J}_k \text{ and } k\in\mathcal{K}_\rho^{\mathrm{x,a}}.
\end{aligned}\tag{3.89}$$

We can use the same observation of (3.59). Thus, since $\lim_{\substack{k\to+\infty \\ k\in\mathcal{K}_\rho^{\mathbf{x},\mathbf{a}}}} \mathbf{x}_k = \mathbf{x}^*$, Assumption 3.3 and Proposition 3.3.4 ensure the existence of $\varepsilon > 0$ such that for $k \in \mathcal{K}_\rho^{\mathbf{x},\mathbf{a},\mathbf{d}} \subseteq \mathcal{K}_\rho^{\mathbf{x},\mathbf{a}}$ sufficiently large, $T_{\mathcal{X}}(\mathbf{x}^*) = T_{\mathcal{X}}(\mathbf{x}_k, \varepsilon) = \text{cone}(\mathcal{P}_k \cap T_{\mathcal{X}}(\mathbf{x}_k, \varepsilon)) = \text{cone}(\mathcal{T}^*)$, where we recall that $\mathcal{T}^* = \{\bar{\mathbf{d}}^i\}_{i\in\bar{\mathcal{J}}}$. Taking the limit for $k \to +\infty$ and $k \in \mathcal{K}_\rho^{\mathbf{x},\mathbf{a},\mathbf{d}}$, and using (3.75), (3.81), (3.82), and (3.87), we obtain

$$\left(\sum_{\ell=1}^{m} z_\ell \nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p} z_{m+j} \nabla h_j(\mathbf{x}^*)\right)^\top \bar{\mathbf{d}}^i \geq 0, \quad \forall \bar{\mathbf{d}}^i \in \mathcal{T}^*. \tag{3.90}$$

We recall that $\mathbf{x}^*$ satisfies the EMFCQ conditions. Let $\mathbf{d}$ be the direction satisfying condition (b) of Definition 3.2.3. For every $\mathbf{d} \in T_{\mathcal{X}}(\mathbf{x}^*)$, there exist nonnegative numbers β_i such that

$$\mathbf{d} = \sum_{\bar{\mathbf{d}}^i \in \mathcal{T}^*} \beta_i \bar{\mathbf{d}}^i. \tag{3.91}$$

Thus, from (3.90) and (3.91), since the scalars β_i are nonnegative, we obtain

$$\left(\sum_{\ell=1}^{m} z_\ell \nabla g_\ell(\mathbf{x}^*) + \sum_{j=1}^{p} z_{m+j} \nabla h_j(\mathbf{x}^*)\right)^\top \mathbf{d} \geq 0. \tag{3.92}$$

Considering Definition 3.2.3, the relation (3.92) becomes

$$\sum_{\ell=1}^{m} z_\ell \nabla g_\ell(\mathbf{x}^*)^\top \mathbf{d} \geq 0. \tag{3.93}$$

Theorem 3.3.8 and the definition of z_ℓ for $\ell \in \{1, \dots, m\}$, guarantee

$$z_\ell = 0, \quad \text{for all} \quad \ell \notin \mathcal{I}^+(\mathbf{x}^*). \tag{3.94}$$

Since $\mathbf{x}^*$ satisfies the EMFCQ conditions, (3.93) implies

$$z_\ell = 0, \quad \text{for all } \ell \in \mathcal{I}^+(\mathbf{x}^*). \tag{3.95}$$

Therefore equation (3.90) becomes

$$\left(\sum_{j=1}^{p} z_{m+j} \nabla h_j(\mathbf{x}^*)\right)^\top \mathbf{d}^* \geq 0, \quad \text{for all} \quad \mathbf{d}^* \in \mathcal{T}^*, \tag{3.96}$$

using again Definition 3.2.3 and (3.96), we obtain

$$z_{m+j} = 0, \quad \text{for all} \quad j \in \{1, \dots, p\}. \tag{3.97}$$

In conclusion, we get (3.94), (3.95), and (3.97), contradicting (3.88) and this concludes the proof. □

As stated at the beginning of this section, the same properties can be proved for the sequences generated by `LOG-DFL`. The result is reported in the following theorem as a reference.

Theorem 3.4.7. *Consider the conditions of Theorem 3.2.9. Let $\mathcal{K}_\rho^{\mathbf{x}}$ be the sequence considered in the proof of the theorem, $\mathbf{u}_k^i$ defined as in* (3.32)*, and let*

$$\lambda_\ell(\mathbf{x};\rho) = \frac{\rho}{-g_\ell(\mathbf{x})}, \quad \textit{for all} \quad \ell = 1,\dots,m$$

$$\mu_\ell(\mathbf{x};\rho) = \nu\left(\frac{|h_j(\mathbf{x})|}{\rho}\right)^{\nu-1}, \quad j = 1,\dots,p.$$

Then

$$\lim_{k\to\infty, k\in K}\left|\lambda_\ell(\mathbf{u}_k^i;\rho_k) - \lambda_\ell(\mathbf{x}_k;\rho_k)\right| = 0, \qquad \ell = 1,\dots,m \quad \forall\ i : \mathbf{d}^i \in \bar{T}, \tag{3.98}$$

$$\lim_{k\to\infty, k\in K}\left|\mu_j(\mathbf{u}_k^i;\rho_k) - \mu_j(\mathbf{x}_k;\rho_k)\right| = 0, \qquad j = 1,\dots,q \quad \forall\ i : \mathbf{d}^i \in \bar{T}, \tag{3.99}$$

where we recall $\bar{T} = \mathcal{E} \cap T_{\mathcal{X}}(\bar{\mathbf{x}})$. Furthermore the following sequences are bounded

$$\{\lambda_\ell(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathbf{x}}}, \quad \ell = 1,\dots,m,$$

$$\{\mu_j(\mathbf{x}_k;\rho_k)\}_{k\in\mathcal{K}_\rho^{\mathbf{x}}}, \quad j = 1,\dots,q.$$

3.5 Conclusions

In conclusion, Chapter 3 introduces a robust framework for DFO aimed at solving nonlinearly constrained black-box problems. This framework is embodied in two key methods, `LOG-DFL` and `LOG-DS`, which address distinct yet complementary approaches to constraint management without relying on derivatives — a critical feature for practical applications.

`LOG-DFL`, the first method presented, is centered on a mixed penalty-barrier merit function tailored for constrained black-box problems. A defining characteristic of `LOG-DFL` is its management of non-relaxable inequality constraints through a log-barrier penalization approach, making it particularly well-suited for tightly constrained problems. The main objective of this work was to demonstrate that the interior penalization of inequality constraints can be efficiently introduced in a DFO setting. The `LOG-DFL` algorithm achieves this through three core components: the interior penalization approach, a DFO line-search exploration, and an automatic rule for updating the penalty-barrier parameter. Together, these elements ensure convergence to stationary points under standard assumptions, which notably do not require any convexity on the problem functions.

An important feature of `LOG-DFL`'s theoretical contribution is its applicability to problems containing both equality and inequality constraints, a versatility that distinguishes it from similar solvers, such as `NOMAD`, where such general constraint handling may be less supported. Additionally, numerical results indicate that `LOG-DFL` is both robust and efficient when benchmarked against `NOMAD` on a wide array of problems from the CUTEst test suite. The encouraging performance shown in these empirical tests underscores `LOG-DFL`'s practical utility. `LOG-DFL` is also openly accessible via the DFL library as the package `LOGDFL`, downloadable at https://github.com/DerivativeFreeLibrary/.

`LOG-DS` builds on the foundational ideas of `LOG-DFL`, extending them to address broader constraints through a GPS framework. By adapting the `SID-PSM` algorithm, which leverages polynomial models at both search and poll steps to improve numerical performance, `LOG-DS` enhances the flexibility of constraint handling and broadens the scope to include general nonlinear and linear constraints. `LOG-DS` introduces a mixed penalty-logarithmic barrier merit function that effectively separates inequality constraints, employing exterior penalties for certain constraints and a logarithmic barrier for others. This division enables `LOG-DS` to dynamically manage constraints as the solution progresses, a feature that allows `LOG-DS` to more effectively navigate feasible regions within complex constraint landscapes.

The theoretical contribution of `LOG-DS` is underscored by its convergence properties, which are established under standard assumptions without, again, the need for convexity in the problem functions. This flexibility ensures `LOG-DS`'s adaptability to a wide range of constraint-intensive problems, making it a powerful tool in constraint-rich BBO. Surrogate models further contribute to `LOG-DS`'s performance by providing efficient merit function approximations that limit unnecessary evaluations, thus optimizing computational load and accelerating convergence. Numerical experiments validate the effectiveness of `LOG-DS`, showcasing its robustness and efficiency across various benchmarks from the CUTEst collection and consistently outperforming other state-of-the-art solvers in accuracy and function evaluation reduction.

Together, `LOG-DFL` and `LOG-DS` contribute a significant advancement to the theoretical and practical landscape of DFO. By establishing convergence without derivative reliance, incorporating complex constraint management techniques, and demonstrating empirical efficacy across numerous test cases, these methods provide a solid foundation for future research. Promising directions for extending this work include adapting the MPB approach for non-smooth DFO problems and exploring its potential in multiobjective DFO contexts.

Bibliography

[1] M. A. Abramson, O. A. Brezhneva, J. E. D. Jr., and R. L. Pingel, *Pattern search in the presence of degenerate linear constraints*, Optim. Methods Softw., 23 (2008), pp. 297–319.

[2] V. S. Amaral, R. Andreani, E. G. Birgin, D. S. Marcondes, and J. M. Martínez, *On complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization*, Journal of Global Optimization, (2022), pp. 1–35.

[3] C. Audet, *A Survey on Direct Search Methods for Blackbox Optimization and Their Applications*, Springer New York, 2014, pp. 31–56.

[4] C. Audet and J. E. Dennis Jr, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM Journal on optimization, 17 (2006), pp. 188–217.

[5] C. Audet, K. J. Dzahini, M. Kokkolaras, and S. Le Digabel, *Stochastic mesh adaptive direct search for blackbox optimization using probabilistic estimates*, Computational Optimization and Applications, 79 (2021), pp. 1–34.

[6] C. Audet and W. L. Hare, *Derivative-free and blackbox optimization*, Springer Ser. Oper. Res. Financ. Eng., Springer, Cham, 2017.

[7] C. Audet and J. E. D. Jr., *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2003), pp. 889–903.

[8] ——, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.

[9] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes, *The NOMAD project.*

[10] H. H. Bauschke and J. M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Review, 38 (1996), pp. 367–426.

[11] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.

[12] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1982.

[13] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific Belmont, MA, 2016.

[14] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numerica, 4 (1995), p. 1–51.

[15] I. M. Bomze, F. Rinaldi, and S. R. Bulò, *First-order methods for the impatient: Support identification in finite time with convergent Frank–Wolfe variants*, SIAM J. Optim., 29 (2019), pp. 2211–2226.

[16] I. M. Bomze, F. Rinaldi, and D. Zeffiro, *Active set complexity of the away-step Frank-Wolfe algorithm*, SIAM J. Optim., 30 (2020), pp. 2470–2500.

[17] D. M. Bortz and C. T. Kelley, *The simplex gradient and noisy optimization problems*, in Computational Methods in Optimal Design and Control, Progress in Systems and Control Theory, J. T. Borggaard, J. Burns, E. Cliff, and S. Schreck, eds., vol. 24, Birkhäuser, Boston, 1998, pp. 77–90.

[18] A. Brilli, A. Cristofari, G. Liuzzi, and S. Lucidi, *Complexity results and active-set identification of a derivative-free method for bound-constrained problems*, 2024.

[19] A. Brilli, A. L. Custódio, G. Liuzzi, and E. J. Silva, *Nonlinear derivative-free constrained optimization with a mixed penalty-logarithmic barrier approach and direct search*, 2024.

[20] A. Brilli, M. Kimiaiei, G. Liuzzi, and S. Lucidi, *Worst case complexity bounds for linesearch-type derivative-free algorithms*, Journal of Optimization Theory and Applications, (2024).

[21] A. Brilli, G. Liuzzi, and S. Lucidi, *An interior point method for nonlinear constrained derivative-free optimization*, 2021.

[22] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[23] E. F. Campana, G. Liuzzi, S. Lucidi, D. Peri, V. Piccialli, and A. Pinto, *New global optimization methods for ship design problems*, Optimization and Engineering, 10 (2009), pp. 533–555.

[24] C. Cartis, N. I. M. Gould, and P. L. Toint, *On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization*, SIAM Journal on Optimization, 22 (2012), pp. 66–86.

[25] C. Cartis, N. I. M. Gould, and P. L. Toint, *Evaluation complexity of algorithms for nonconvex optimization: Theory, computation and perspectives*, 2022.

[26] R. Chen, M. Menickelly, and K. Scheinberg, *Stochastic optimization using a trust-region method and random models*, Mathematical Programming, 169 (2018), pp. 447–487.

[27] P. Conejo, E. W. Karas, L. G. Pedroso, A. A. Ribeiro, and M. Sachine, *Global convergence of trust-region algorithms for convex constrained minimization without derivatives*, Appl. Math. and Comput., 220 (2013), pp. 324–330.

[28] A. R. Conn, N. Gould, A. Sartenaer, and P. L. Toint, *Convergence Properties of an Augmented Lagrangian Algorithm for Optimization with a Combination of General Equality and Linear Constraints*, SIAM J. Optim., 6 (1996), pp. 674–703.

[29] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust-Region Methods*, SIAM, 2000.

[30] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, SIAM, 2009.

[31] I. Coope and C. Price, *Positive bases in numerical optimization*, Computational Optimization and Applications, 21 (2002).

[32] A. Cristofari, *Active-set identification with complexity guarantees of an almost cyclic 2-coordinate descent method with Armijo line search*, SIAM J. Optim., 32 (2022), pp. 739–764.

[33] A. Cristofari and F. Rinaldi, *A derivative-free method for structured optimization problems*, SIAM J. Optim., 31 (2021), pp. 1079–1107.

[34] F. E. Curtis, *A penalty-interior-point algorithm for nonlinear constrained optimization*, Mathematical Programming Computation, 4 (2012), pp. 181–209.

[35] A. L. Custódio, H. Rocha, and L. N. Vicente, *Incorporating minimum frobenius norm models in direct search*, Comput. Optim. Appl., 46 (2010), pp. 265–278.

[36] A. L. Custódio and L. N. Vicente, *Using sampling and simplex derivatives in pattern search methods*, SIAM J. Optim., 18 (2007), pp. 537–555.

[37] C. Davis, *Theory of positive linear dependence*, American Journal of Mathematics, 76 (1954).

[38] R. De Leone, M. Gaudioso, and L. Grippo, *Stopping criteria for linesearch methods without derivatives*, Mathematical Programming, (1984). Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[39] M. Dodangeh and L. N. Vicente, *Worst case complexity of direct search under convexity*, Math. Program., 155 (2016), pp. 307–332.

[40] M. Dodangeh, L. N. Vicente, and Z. Zhang, *On the optimal order of worst case complexity of direct search*, Optimization Letters, 10 (2016), p. 699–708.

[41] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[42] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi, *A linesearch-based derivative-free approach for nonsmooth constrained optimization*, SIAM journal on optimization, 24 (2014), pp. 959–992.

[43] Y. Feng, O. el Moctar, and T. E. Schellin, *Parametric hull form optimization of containerships for minimum resistance in calm water and in waves*, Journal of Marine Science and Application, 20 (2021), pp. 670–693.

[44] E. Fermi and N. Metropolis, *Numerical solution of a minimum problem*, Tech. Rep. Los Alamos Unclassified Report LS–1492, Alamos National Laboratory, Los Alamos, NM, 1952.

[45] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*, SIAM, 1990.

[46] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Review, 44 (2002), pp. 525–597.

[47] R. Garmanjani, D. Júdice, and L. N. Vicente, *Trust-Region Methods Without Using Derivatives: Worst Case Complexity and the NonSmooth Case*, SIAM J. Optim., 26 (2016), pp. 1987–2011.

[48] R. Garmanjani and L. N. Vicente, *Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization*, IMA Journal of Numerical Analysis, 33 (2012), pp. 1008–1028.

[49] N. I. Gould, D. Orban, and P. L. Toint, *Cutest: a constrained and unconstrained testing environment with safe threads for mathematical optimization*, Computational optimization and applications, 60 (2015), pp. 545–557.

[50] G. N. Grapiglia, *Quadratic regularization methods with finite-difference gradient approximations*, Computational Optimization and Application, 85 (2023).

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

[52] S. Gratton, P. L. Toint, and A. Tröltzsch, *An active-set trust-region method for derivative-free nonlinear bound-constrained optimization*, Optim. Methods Softw., 26 (2011), pp. 873–894.

[53] L. Grippo, F. Lampariello, and S. Lucidi, *Global convergence and stabilization of unconstrained minimization methods without derivatives*, Journal of Optimization Theory and Applications, 56 (1988), pp. 385–406.

[54] E. A. Gumma, M. Hashim, and M. M. Ali, *A derivative-free algorithm for linearly constrained optimization problems*, Comput. Optim. Appl., 57 (2014), pp. 599–621.

[55] W. Hare and G. Jarry-Bolduc, *A deterministic algorithm to compute the cosine measure of a finite positive spanning set*, Optimization Letters, 14 (2020).

[56] W. Hare, G. Jarry-Bolduc, and C. Planiden, *Nicely structured positive bases with maximal cosine measure*, Optimization Letters, 17 (2023).

[57] W. L. Hare and A. S. Lewis, *Identifying active constraints via partial smoothness and prox-regularity*, J. Convex Anal., 11 (2004), pp. 251–266.

[58] R. Hooke and T. A. Jeeves, *“ direct search” solution of numerical and statistical problems*, J. ACM, 8 (1961).

[59] M. Hough and L. Roberts, *Model-Based Derivative-Free Methods for Convex-Constrained Optimization*, SIAM J. Optim., 32 (2022), pp. 2552–2579.

[60] T. Kolda, R. Lewis, and V. Torczon, *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Review, 45 (2003), pp. 385–482.

[61] T. G. Kolda, R. M. Lewis, and V. Torczon, *Stationarity results for generating set search for linearly constrained optimization*, SIAM J. Optim., 17 (2007), pp. 943–968.

[62] J. Larson and S. C. Billups, *Stochastic derivative-free optimization using a trust region framework*, Computational Optimization and applications, 64 (2016), pp. 619–645.

[63] J. Larson, M. Menickelly, and S. M. Wild, *Derivative-free optimization methods*, Acta Numerica, 28 (2019), p. 287–404.

[64] S. Le Digabel and S. M. Wild, *A taxonomy of constraints in black-box simulation-based optimization*, Optimization and Engineering, (2023), pp. 1–19.

[65] R. M. Lewis and V. Torczon, *Pattern Search Algorithms for Bound Constrained Minimization*, SIAM J. Optim., 9 (1999), pp. 1082–1099.

[66] R. M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM J. Optim., 10 (2000), pp. 917–941.

[67] R. M. Lewis and V. Torczon, *A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds*, SIAM Journal on Optimization, 12 (2002), pp. 1075–1089.

[68] ——, *Active set identification for linearly constrained minimization without explicit derivatives*, SIAM J. Optim., 20 (2010), pp. 1378–1405.

[69] R. M. Lewis, V. Torczon, and M. W. Trosset, *Direct search methods: then and now*, Journal of Computational and Applied Mathematics, 124 (2000), pp. 191–207. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations.

[70] C.-J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, *Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds*, Journal of Optimization Theory and Applications, 141 (2009), pp. 107–126.

[71] G. Liuzzi and S. Lucidi, *A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function*, SIAM Journal on Optimization, 20 (2009), pp. 1–29.

[72] G. Liuzzi, S. Lucidi, and M. Sciandrone, *A derivative-free algorithm for linearly constrained finite minimax problems*, SIAM J. Optim., 16 (2006), pp. 1054–1075.

[73] G. Liuzzi, S. Lucidi, and M. Sciandrone, *Sequential penalty derivative-free methods for nonlinear constrained optimization*, SIAM Journal on Optimization, 20 (2010), pp. 2614–2635.

[74] S. Lucidi and M. Sciandrone, *A derivative-free algorithm for bound constrained optimization*, Computational Optimization and Applications, 21 (2002), pp. 119–142.

[75] S. Lucidi and M. Sciandrone, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM Journal on Optimization, 13 (2002), pp. 97–116.

[76] S. Lucidi, M. Sciandrone, and P. Tseng, *Objective-derivative-free methods for constrained optimization*, Math. Program., 92 (2002), pp. 37–59.

[77] O. L. Mangasarian and S. Fromovitz, *The fritz john necessary optimality conditions in the presence of equality and inequality constraints*, Journal of Mathematical Analysis and Applications, 17 (1967), pp. 37–47.

[78] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 20 (2009), pp. 172–191.

[79] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

[80] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[81] J. Nutini, M. Schmidt, and W. L. Hare, *"Active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern?*, Optim. Lett., 13 (2019), pp. 645–655.

[82] E. Polak, ed., *Computational Methods in Optimization: A Unified Approach*, vol. 77 of Mathematics in Science and Engineering, Elsevier, 1971.

[83] M. J. Powell, *On fast trust region methods for quadratic models with linear constraints*, Math. Program. Comput., 7 (2015), pp. 237–267.

[84] T. J. R. Hooke, *Direct search solution of numerical and statistical problems*, J. Assoc. Comput. Mach., (1961).

[85] V. Torczon, *On the convergence of pattern search algorithms*, SIAM Journal on Optimization, 7 (1997), pp. 1–25.

[86] L. N. Vicente, *Worst case complexity of direct search*, EURO Journal on Computational Optimization, 1 (2013), pp. 143–153.

[87] S. J. Wright, *Identifiable surfaces in constrained optimization*, SIAM J. Control Optim., 31 (1993), pp. 1063–1079.