NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

CONSTRAINED MULTIOBJECTIVE DERIVATIVE-FREE OPTIMIZATION

EVERTON JOSÉ DA SILVA

Master in Mathematics

DOCTORATE IN MATHEMATICS

NOVA University of Lisbon
April, 2025

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

CONSTRAINED MULTIOBJECTIVE DERIVATIVE-FREE OPTIMIZATION

EVERTON JOSÉ DA SILVA

Master in Mathematics

Adviser: Ana Luísa da Graça Batista Custódio
Associate Professor, NOVA School of Science and Technology, NOVA University of Lisbon

Examination Committee

Chair: João Jorge Ribeiro Soares Gonçalves de Araújo
Full Professor, NOVA School of Science and Technology, NOVA University of Lisbon

Rapporteurs: Francesco Rinaldi
Full Professor, University of Padova

José Firmino Aguilar Madeira
Coordinator Professor with Habilitation, Instituto Superior de Engenharia de Lisboa

Adviser: Ana Luísa da Graça Batista Custódio
Associate Professor, NOVA School of Science and Technology, NOVA University of Lisbon

Members: João Jorge Ribeiro Soares Gonçalves de Araújo
Full Professor, NOVA School of Science and Technology, NOVA University of Lisbon

Maria do Carmo Proença Caseiro Brás
Associate Professor, NOVA School of Science and Technology, NOVA University of Lisbony

Marcos Alejandro Raydan
Principal Researcher, NOVA School of Science and Technology, NOVA University of Lisbon

Rohollah Garmanjani
Assistant Researcher, NOVA School of Science and Technology, NOVA University of Lisbon

DOCTORATE IN MATHEMATICS

NOVA University of Lisbon
April, 2025

Constrained Multiobjective Derivative-free Optimization

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.10) [137].

Dedico esta dissertação à minha mãe, Maria.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to the Department of Mathematics at NOVA School of Science and Technology for providing a stimulating academic environment and essential support throughout my Ph.D. journey. I also extend my appreciation to the Ph.D. Program in Mathematics, which offered a solid education and a nurturing environment for scientific growth during the development of this work. Furthermore, I am deeply grateful to the Center for Mathematics and Applications (NOVA Math) for its funding and institutional support, which enabled my research and facilitated the exchange of knowledge within a highly qualified scientific community.

The financial support provided by Fundação para a Ciência e a Tecnologia (FCT) deserves special recognition. Thanks to this funding, I was able to conduct this research and complete this thesis under project UI/BD/151246/2021. I also acknowledge the support received through the project BoostDFO: Improving the Performance and Exploring New Directions in Derivative-Free Optimization (PTDC/MAT-APL/28400/2017).

I would like to express my deepest gratitude to everyone whose knowledge and experience have played a fundamental role in my academic growth. Every lesson learned, every challenge overcame, and every exchange of ideas has contributed immensely to my journey. The impact of their guidance and support extends beyond academic learning—it has shaped my worldview and way of thinking, and I will carry it with me forever. I am eternally grateful for everything I have learned and for every step taken alongside such inspiring individuals. In particular, I thank the professors of the Ph.D. program, who have been steadfast supporters of my personal and professional development.

I express my sincere gratitude to Giampaolo Liuzzi, from La Sapienza University of Rome, for his valuable collaboration and insightful discussions throughout this journey. His knowledge, perspectives, and dedication have been fundamental in developing new ideas and approaches in our research. Working alongside him has been an enriching experience, and I am deeply grateful for the exchange of knowledge and his support.

My thanks to the Thesis Accompanying Committee (CAT), Marcos Raydan and Gabriele Eichfelder, for their valuable contributions, which were essential to the successful completion of this project.

I would like to express my deepest gratitude to my Ph.D. advisor, Ana Luisa Custódio, for her guidance throughout this work. I am immensely grateful for her dedication, suggestions, corrections, patience, and, above all, for the friendship we built over the years of study.

I extend my heartfelt thanks to my friends at the Center for Mathematics and Applications (NOVA Math) for making the university an even more enjoyable place through the exchange of knowledge, friendship, and camaraderie. A special thanks to Roberta, Bárbara, and Nádia for the moments and experiences we shared, which I will cherish for the rest of my life.

A special thanks to Andrea, whose collaboration has been a constant source of learning and motivation throughout this academic journey. Since we started working together, our partnership has been marked by enthusiasm in exploring new research directions and by sharing challenges and achievements. The exchange of ideas and academic camaraderie has made this path significantly more enriching and rewarding. I am deeply grateful for this experience and for the privilege of sharing this journey with someone so dedicated and inspiring.

I would like to express my immense gratitude to my family for their unconditional support, encouragement, and love throughout this journey. The trust you have always placed in me has been a constant source of strength, and without you, I would not have reached this stage.

To Henrique, for his patience and care, for encouraging me to persevere even in the face of difficulties, and, above all, for believing in me. Together, we have overcome challenges that life has placed before us and have grown stronger. I am immensely grateful for your unwavering support.

I am deeply grateful to my dear friend Flávia, who stood by my side during moments of difficulty, doubt, and apprehension, by offering unwavering support and friendship, and to my friend and housemate Bruno, whose companionship and understanding enriched both joyful moments and challenging times during our years together in Almada.

To my dear friends Milena, Rodrigo, and Fernando, I am deeply grateful for every moment we shared, for the laughter, and for the memories we created together. Each one of you has made my journey even more special and unforgettable.

Finally, I extend my heartfelt gratitude to all my friends from the Fraústo da Silva residency, where I experienced unforgettable moments for almost two years. During that time, I had the privilege of meeting incredible people from different parts of the world—including Turkey, Italy, China, Moldova, Ukraine, Angola, Mozambique, Nigeria, Spain, Russia, Morocco, Iran, India, Pakistan, Poland, Greece, and Germany—each of whom has left a special mark on my life. I am grateful for all the meaningful exchanges, experiences, and friendships we built, and I hope our connection endures regardless of the distance.

"We can only see a short distance ahead, but we can see plenty there that needs to be done."
(Alan Turing)

Abstract

The present work focus on multiobjective derivative-free optimization, proposing strategies to address constrained problems.

For that, a direct multisearch method that combines a filter-based strategy with an inexact restoration phase, DMS-FILTER-IR, was developed. In this approach, feasibility is treated as an additional component of the objective function that must be minimized. The inexact restoration step attempts to generate new feasible points, thereby prioritizing feasibility, a requirement for the strong performance of any filter approach. Theoretical results are provided, analyzing the different types of sequences of points generated by the new algorithm, and numerical experiments on a set of nonlinearly constrained biobjective problems are reported, stating the good algorithmic performance of the proposed approach.

The filter approach reformulates the original problem by aggregating the constraint violations into an additional objective function component, thereby increasing the number of objectives by one. From a theoretical point of view, DMS-FILTER-IR was developed for continuous constrained multiobjective optimization with an arbitrary number of objectives. However, when the original problem has three or more objectives, the increase of this number caused by the use of the filter, originates a many-objective optimization problem. Thus, strategies to address many-objective optimization problems are also investigated.

Based on reduction approaches, employing correlation or sketching techniques, we propose a new variant of DMS, namely DMS-Reduction. This reduction method aims to tackle large problems by decreasing both the number of objective function components and the number of problem variables. Reducing the number of components of the objective function to be optimized at each iteration, has the additional benefit of potentially conducting to a reduction in the number of variables to be optimized, since there could be the case that not all variables are related to the objective function components selected. We detail the proposed algorithm and report a large set of numerical experiments that demonstrate the potential of this approach in addressing many-objective optimization problems.

A different way of addressing constraints is resorting to penalization techniques. We

investigate the use of a logarithmic barrier technique, both in single and multiobjective derivative-free optimization. For single-objective optimization, we propose the joint use of a mixed penalty-logarithmic barrier method and direct search. A merit function is employed in which the set of inequality constraints is divided into two groups: one is treated with a logarithmic barrier approach, while the other, together with the equality constraints, is addressed using a penalization term. This strategy is incorporated into a generalized pattern search method, enabling the effective handling of general nonlinear constraints. Convergence to KKT-stationary points is established under continuous differentiability assumptions, without requiring any form of convexity. Numerical experiments demonstrate the robustness, efficiency, and overall effectiveness of the proposed method, when compared with state-of-the-art solvers. The logarithmic barrier approach is then extended to the multiobjective setting via LOG-DMS, demonstrating improved exploration of the Pareto front under inequality constraints. Comparative experiments indicate that LOG-DMS outperforms traditional DMS and DMS-FILTER-IR in terms of hypervolume metrics.

Overall, the contributions of this thesis not only advance the theoretical understanding of constrained derivative-free optimization methods but also provide practical, competitive algorithms for solving optimization problems.

Keywords: Derivative-free optimization, Black-box optimization, Multiobjective optimization, Constrained optimization, Direct multisearch, Filter methods, Inexact restoration, Many-objective optimization, Interior point methods, Direct search, Pareto front

Resumo

O presente trabalho centra-se na otimização multiobjetivo sem derivadas, propondo estratégias para abordar problemas com restrições.

Para tal, foi desenvolvido um método de pesquisa múltipla direta, DMS-FILTER-IR, que combina uma estratégia baseada em filtro com uma fase de restauração inexata. Nesta abordagem, a viabilidade é tratada como uma componente adicional da função objetivo que deve ser minimizada. A etapa de restauração inexata procura gerar novos pontos admissíveis, priorizando assim a admissibilidade, um requisito para o forte desempenho de qualquer abordagem de filtro. São apresentados resultados teóricos, analisando os diferentes tipos de sequências de pontos geradas pelo novo algoritmo, e são relatados experimentos numéricos num conjunto de problemas biobjetivo com restrições não lineares, evidenciando o bom desempenho algorítmico da abordagem proposta.

A abordagem baseada em filtro reformula o problema original ao agregar as violações das restrições em uma componente adicional da função objetivo, aumentando assim o número de objetivos em um. Do ponto de vista teórico, o DMS-FILTER-IR foi desenvolvido para a otimização multiobjetivo contínua com restrições, com um número arbitrário de objetivos. No entanto, quando o problema original tem três ou mais objetivos, o aumento deste número causado pelo uso do filtro origina um problema de otimização com muitos objetivos. Assim, estratégias para abordar problemas de otimização com muitos objetivos também são investigadas.

Com base em abordagens de redução, empregando técnicas de correlação ou esboço, propomos uma nova variante do DMS, nomeadamente o DMS-Reduction. Este método de redução tem como objetivo resolver problemas de grande dimensão, diminuindo tanto o número de componentes da função objetivo quanto o número de variáveis do problema. Reduzir o número de componentes da função objetivo a ser otimizado em cada iteração tem o benefício adicional de, potencialmente, conduzir a uma redução no número de variáveis a serem otimizadas, uma vez que pode acontecer que nem todas as variáveis estejam relacionadas com os componentes da função objetivo selecionados. Detalhamos o algoritmo proposto e relatamos um vasto conjunto de experimentos numéricos que demonstram o potencial desta abordagem na resolução de problemas de otimização com

muitos objetivos.

Uma forma diferente de tratar as restrições é recorrer a técnicas de penalização. Investigamos o uso de uma técnica de barreira logarítmica, tanto na otimização sem derivadas de objetivo único quanto multiobjetivo. Para a otimização com um único objetivo, propomos o uso conjunto de um método misto de penalização e barreira logarítmica com a pesquisa direta. Utiliza-se uma função mérito na qual o conjunto de restrições de desigualdade é dividido em dois grupos: um é tratado com uma abordagem de barreira logarítmica, enquanto o outro, juntamente com as restrições de igualdade, é abordado através de um termo de penalização. Esta estratégia é incorporada num método de pesquisa em padrão generalizada, permitindo o tratamento eficaz de restrições não lineares. A convergência para pontos estacionários de KKT é estabelecida sob pressupostos de diferenciabilidade contínua, sem exigir qualquer forma de convexidade. Experimentos numéricos demonstram a robustez, eficiência e eficácia global do método proposto, quando comparado com solucionadores de última geração. A abordagem de barreira logarítmica é, a seguir, estendida ao contexto multiobjetivo através do LOG-DMS, demonstrando uma exploração melhorada da frente de Pareto sob restrições de desigualdade. Experimentos comparativos indicam que o LOG-DMS supera o DMS tradicional e o DMS-FILTER-IR em termos de métricas de hipervolume.

De forma geral, as contribuições desta tese não só avançam a compreensão teórica dos métodos de otimização sem derivadas com restrições, como também fornecem algoritmos práticos e competitivos para a resolução de problemas de otimização.

Palavras-chave: Otimização sem derivadas, Otimização de caixa preta, Otimização multiobjetivo, Otimização com restrições, Procura direta múltipla, Métodos de filtro, Restauração inexata, Otimização de muitos objetivos, Métodos de pontos interiores, Procura direta, Frente de Pareto

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

ALAMO	Augmented Lagrangian for Multiobjective Optimization
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BiMADS	Biobjective Mesh Adaptive Direct Search
BoostDMS	Boost Direct Multisearch
CQ	Constraint Qualification
CRCQ	Constant Rank Constraint Qualification
DDS	Directional Direct Search
DFMO	Derivative-Free Multiobjective Optimization
DFO	Derivative-free Optimization
DMS	Direct Multisearch
DMS-FILTER-IR	Direct Multisearch Filter approach with Inexact Restoration step
DMS-Reduction	Direct Multisearch for many-objective optimization based on reductions
DMultiMADS	Mesh Adaptive Direct Multisearch Algorithm
DMultiMADS-PB	Mesh Adaptive Direct Multisearch Progressive Barrier Algorithm
DMultiMADS-TEB	Mesh Adaptive Direct Multisearch Two-Phase Algorithm
EMFCQ	Extended Mangasarian Fromovitz Constraint Qualification
GCQ	Guignard Constraint Qualification
GPS	Generalized Pattern Search
GrEA	Grid-Based Evolutionary Algorithm
KKT	Karush-Kuhn-Tucker
KnEA	Knee point driven Evolutionary Algorithm

L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
LICQ	Linear Independence Constraint Qualification
LOG-DMS	Logarithmic Barrier in Direct Multisearch
LOG-DS	Logarithmic Barrier in Direct Search
LTMADS	Mesh Adaptive Direct Search, using a Lower Triangular basis approach
MADS	Mesh Adaptive Direct Search
MFCQ	Mangasarian Fromovitz Constraint Qualification
MOEA/DD	Many-Objective Evolutionary Algorithm based on Dominance and-Decomposition
MOMBI-II	Many-Objective Metaheuristic based on the R2 indicator II
MOO	Multiobjective Optimization
MULTIMADS	Multiobjective Mesh Adaptive Direct Search
NBI	Normal Boundary Intersection
NOMAD	Nonlinear Optimization with the MADS Algorithm
NSGA-III	Nondominated Sorting Genetic Algorithm
ORTHOMADS	Mesh Adaptive Direct Search with Orthogonal directions
Parallel BoostDMS	parallel version of BoostDMS
PlatEMO	Evolutionary Multiobjective Optimization Platform
SID-PSM	Pattern Search Method based on Simplex Derivatives
SMS	Sequential Modular Simulation
SQP	Sequential Quadratic Programming
X-LOG-DFL	Logarithmic Barrier Derivative-Free Linesearch

Symbols

$B(\overline{x}, \delta)$	Open ball centered at $\overline{x}$ with radius δ
$f^{\circ}(\boldsymbol{x}; \boldsymbol{d})$	The generalized directional derivative computed at $\boldsymbol{x}$ along the direction $\boldsymbol{d}$
$\partial f(\boldsymbol{x})$	The Clarke subdifferential
$T_{\Omega}^{Cl}(\boldsymbol{x})$	The Clarke tangent cone to Ω computed at $\boldsymbol{x}$
$\mathrm{cl}(\Omega)$	The closure of the Ω
$\mathrm{cone}(\boldsymbol{D}_k \cap T_X(\boldsymbol{x}_k, \varepsilon))$	The cone generated by vectors in $\boldsymbol{D}_k \cap T_X(\boldsymbol{x}_k, \varepsilon)$
$\nabla f(\boldsymbol{x})$	The gradient of function f computed at point $\boldsymbol{x}$
$\nabla^2 f(\boldsymbol{x}_k)$	The Hessian of function f computed at point $\boldsymbol{x}_k$
$H_{\Omega}^{Cl}(\boldsymbol{x})$	The Clarke hypertangent cone to Ω computed at $\boldsymbol{x}$
$I(\boldsymbol{x})$	The set of indices of active inequalities at a feasible point $\boldsymbol{x} \in \Omega$
$\mathrm{int}(T_{\Omega}^{Cl}(\boldsymbol{x}))$	The interior of set $T_{\Omega}^{Cl}(\boldsymbol{x})$
$J_{\mathbf{F}}(\cdot)$	The Jacobian matrix of $\mathbf{F}$
$L_{\Omega}(\boldsymbol{x})$	The linearizing cone of Ω around $\boldsymbol{x}$
$\mathbb{N}$	The set of natural numbers
$\|\cdot\|$	The Euclidean norm for vectors or operator 2-norm for matrices
$(\cdot)^{\circ}$	The polar cone of a set
$\mathbb{R}$	The set of real numbers
$\mathbb{R}_+$	The set of nonnegative real numbers

$T_\Omega(\boldsymbol{x})$	The tangent cone to Ω computed at $\boldsymbol{x}$
$\mathcal{V}(\cdot)$	Lebesgue measure of a set of points
$\mathbb{Z}$	The set of integer numbers

1

Introduction

This chapter offers a broad introduction to the thesis, establishing its context and objectives to provide readers with a clear foundation for what follows.

It begins in Section 1.1 with a concise introduction to optimization, outlining key fundamental concepts. Section 1.2 defines the research problem, detailing the specific challenges the thesis aims to address. The main contributions of the work are presented in Section 1.3, emphasizing the novel insights and advancements achieved. Lastly, Section 1.4 describes the thesis structure, offering an overview of the organization and content of the subsequent chapters.

1.1 A brief introduction to optimization

Optimization is the mathematical process of finding the best solution among a set of feasible alternatives. At its core, an optimization problem involves either minimizing or maximizing an objective function, often subject to a set of constraints. In unconstrained optimization, no restrictions are imposed, so the goal is to identify the best solution over the entire variable domain. Conversely, constrained optimization requires that the solution satisfy specific conditions that may be expressed as equalities, inequalities, or a combination of both, and can be formulated using either linear or nonlinear mathematical expressions. As a result, constrained optimization problems are inherently more complex than unconstrained ones, due to the added requirement of meeting these conditions, which increases the theoretical and computational challenges involved.

Constraints play a crucial role in real-world optimization problems, as they reflect the inherent limitations and complexities of practical situations. For instance, in engineering, design parameters are bound by physical limitations, while in finance, investment strategies must navigate budget constraints – including nonlinear constraints related to risk measures, like variance – reflecting the complex dynamics of financial markets. Constrained optimization emerges as a valuable tool for incorporating these real-world limitations into the problem-solving process, ensuring practical and viable solutions.

Handling general constraints in optimization problems presents several difficulties.

The presence of constraints can transform a simple optimization problem into a complex challenge, as feasible solutions must satisfy all imposed requirements. This can lead to a very small feasible region, making it difficult to identify solutions that are both feasible and optimal. Advanced constraint-handling methods, such as penalty functions, barrier methods, or augmented Lagrangian approaches, are often employed to address these issues. The choice of method typically depends on the nature of the constraints and the structure of the problem.

Derivatives play a crucial role in nonlinear optimization, as they can guide the search by identifying potential descent directions for the functions. Nonetheless, there are various reasons why derivatives might not be available. For example, the problem itself may be nonsmooth or, even if smooth, could be affected by numerical noise or function evaluation could be computationally expensive. In such cases, alternative methods, such as derivative-free optimization techniques, become essential.

Derivative-free optimization problems are frequently linked to black-box functions, particularly in the context of simulation-based optimization [14, 60]. In such cases, function evaluation involves running a typically costly numerical simulation, which, given some input corresponding to the problem variables, only returns a vector of function values corresponding to the objective and constraints.

Another key distinction is between single-objective and multiobjective optimization. In single-objective optimization, the objective function is a scalar. Thus, the optimal solution is often a single point. In contrast, multiobjective optimization deals with multiple, often conflicting, objectives, resulting in a set of trade-off solutions known as Pareto optimal or nondominated points.

Conventional strategies for tackling multiobjective optimization problems often rely on scalarization techniques [80, 144], which transform the multiobjective problem into a single-objective one. This allows the solutions to the multiobjective problem to be obtained as solutions to a classical nonlinear single-objective optimization problem. However, scalarization requires careful selection of weights, which can greatly influence the trade-offs between objectives. In practice, choosing these weights may involve expert knowledge, sensitivity analysis, or adaptive schemes designed to better capture the decision-maker's preferences. Moreover, since each scalarization run typically produces only one solution, multiple runs are necessary to approximate the full Pareto front, which can be computationally demanding.

An alternative approach involves the use of evolutionary algorithms [70] or other meta-heuristic methods [22]. In the former, a population of candidates solutions is iteratively improved through crossover, mutation, and selection operators. Despite their flexibility and ease of implementation, these approaches do not scale well and lack theoretical guarantees supporting the numerical performance.

More recently, considerable research effort has been dedicated to extending iterative techniques initially designed for single-objective optimization to multiobjective optimization, offering a viable alternative to both scalarization approaches and meta-heuristics.

In the last decade, classes of derivative-free optimization methods have been developed and analyzed for multiobjective optimization. In fact, in a recent survey [126], multiobjective derivative-free optimization was considered as "an especially open avenue of future research".

1.2 Problem statement

In this thesis, we consider the general constrained multiobjective derivative-free optimization problem given by

$$\begin{aligned} &\min \ \mathbf{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^\top \\ &\text{s. t. } \boldsymbol{x} \in \Omega, \end{aligned}$$

where $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$, with $m \in \mathbb{N}$ with $m \geq 2$ and $\Omega \subseteq \mathbb{R}^n$ is the feasible set.

The presence of several conflicting objectives that need to be simultaneously optimized, changes the classical concept of problem solution, which is no longer a single point. Instead, the solution set consists of trade-off solutions known as nondominated points. These points represent scenarios where improving one objective is only possible at the expense of another, highlighting the inherent conflicts among the objectives. The image, by the objective function, of this set of nondominated points is denominated as Pareto front and constitutes the solution of the multiobjective optimization problem. It may be supported by a convex, a nonconvex, a connected, or a disconnected function, making its computation difficult [144, 185]. This task is even more complicated if derivatives are not available [14, 60]. In this case, the problem to be solved will be a multiobjective derivative-free optimization problem.

The main goal of this thesis is to develop strategies that explicitly address general constraints in multiobjective derivative-free optimization. Here, we assume that evaluating the objective function is highly expensive. Consequently, a key requirement for the proposed algorithms is their ability to efficiently manage the number of function evaluations, ultimately enhancing the overall performance of the methods. Theoretical properties of these strategies will be analyzed, and their practical effectiveness will be assessed through numerical experiments.

1.3 Contributions of this thesis

This thesis focuses on three key areas within optimization that present significant challenges by themselves: constrained, multiobjective, and derivative-free optimization. This kind of problems demand innovative approaches in the absence of gradient information. The present work makes a valuable contribution by proposing new algorithms specifically designed to handle general constraints, a critical challenge when seeking solutions in real-world problems with restrictive limitations. Often, these algorithms are analyzed

from a theoretical perspective, and their practical effectiveness is validated through comprehensive numerical experiments.

Direct Multisearch (DMS) [64] is a well-established class of methods for multiobjective derivative-free optimization, where constraints are addressed by an extreme barrier approach, only evaluating feasible points. In Chapter 4, we present the results published in [176], where we proposed the replacement of this extreme barrier approach by a filter strategy, combined with an inexact feasibility restoration step, to address constraints in the DMS.

In the new algorithm, DMS-FILTER-IR, the filter approach treats feasibility as an additional component of the objective function, that needs to be minimized. A proper criterion is defined to decide when to explore feasible or infeasible points at a given iteration (but never both) and the maximum value allowed for constraint violation is never updated. Instead, when the point to be explored at a given iteration is infeasible, the algorithm makes use of an inexact feasibility restoration step. The motivation behind it is that one should not evaluate the possibly expensive objective function, without first trying to restore (or at least improve) feasibility, a requirement for the good performance of any filter approach.

Theoretical results are provided, analyzing the different types of sequences of points generated by the new algorithm, and numerical experiments on a set of nonlinearly constrained biobjective problems are reported, stating the good algorithmic performance of the proposed approach.

Although the theory was established for a general number of objectives, the numerical experiments conducted were only for biobjective problems. Since DMS-FILTER-IR is an algorithm based on a filter approach, a problem initially formulated with only two objectives will now involve three components in the objective function in the new formulation. Therefore, the superior performance of DMS in handling a maximum of either 2 or 3 objectives was also evident in the case of DMS-FILTER-IR.

However, as the number of objectives grows, it becomes much simpler to find new non-dominated points, which poses a challenge due to the proliferation of nondominated solutions. This phenomenon underscores the complexity inherent in many-objective optimization. In Chapter 5, we present a new variant of DMS, named DMS-Reduction, which builds upon a framework relying on search directions chosen within subspaces of the variable space, as proposed in [171], or based on correlation matrices to measure the correlation between variables and objectives.

The reduction is an attempt to tackle large problems, in what respects to the number of components of the objective function and the number of problem variables. By reducing the number of components of the objective function to be optimized at each iteration, we also potentially decrease the number of variables to be optimized, as not all variables may be related to the selected objective function components. Extensive numerical results are reported, showing the potential of DMS-Reduction for many-objective derivative-free optimization.

As an alternative to the filter approach, in Chapter 6, we analyze the use of the logarithmic barrier in direct search methods to address both single-objective and multiobjective derivative-free constrained optimization problems.

Section 6.1 begins with the results currently submitted for publication [38], where we proposed the joint use of a mixed penalty-logarithmic barrier approach and direct search to handle single-objective nonlinearly constrained derivative-free optimization problems.

The proposed strategy relies on a merit function in which the set of inequality constraints is divided into two groups: one treated with a logarithmic barrier approach and another, along with the equality constraints, addressed using a penalization term. Initially introduced in the framework of X-LOG-DFL [39], this approach is adapted and incorporated into the SID-PSM [65, 66] algorithm, a generalized pattern search method.

As a result, LOG-DS has been developed, a direct search method capable of explicitly addressing nonlinear constraints through a mixed penalty-logarithmic barrier approach. Under reasonable smoothness assumptions, convergence is established without requiring any convexity assumptions. Numerical experiments demonstrate the robustness, efficiency, and overall effectiveness of the proposed method compared to state-of-the-art solvers and the original SID-PSM and X-LOG-DFL implementations.

This initial analysis serves as a litmus test to determine whether the logarithmic barrier could be potentially be adapted to a multiobjective setting. Since the single-objective case can be seen as a special instance of the multiobjective framework (with $m = 1$), a lackluster performance in this context would suggest that extending the approach to the multiobjective case might be unproductive. Consequently, only after confirming the efficacy of the barrier method in this simplified scenario we proceeded in Section 6.2 to transpose and adapt the methodology for tackling more complex multiobjective derivative-free constrained optimization problems. A new algorithmic framework, LOG-DMS, is proposed and preliminary numerical experiments are reported, with evidence of the numerical potential of the approach.

In total, the present thesis proposed four new derivative-free algorithms (DMS-FILTER-IR, DMS-Reduction, LOG-DS, and LOG-DMS), two of them with a well-established theoretical analysis, and all with associated numerical experience, that corroborates their potential as relevant optimization tools for the optimization community.

1.4 Organization of this thesis

Chapter 2 lays the mathematical foundation for both single-objective and multiobjective optimization by discussing optimality conditions and related fundamental concepts. Building upon this foundation, Chapter 3 focuses on numerical optimization algorithms for single-objective and multiobjective problems. In this chapter, the algorithms are divided into two broad categories: derivative-based methods, such as linesearch and trust-region techniques, and derivative-free approaches, including direct search strategies

and trust-region methods specifically designed for scenarios where derivative information is unavailable.

Chapter 4 then introduces the novel filter-based direct multisearch approach, with inexact restoration, tailored for multiobjective derivative-fee constrained optimization. This innovative method is designed to efficiently handle constraints by integrating an inexact restoration phase, thereby enhancing the robustness and reliability of the optimization process.

Extending the discussion to many-objective problems, Chapter 5 presents an algorithm that leverages dimension reduction techniques to manage the complexities of high-dimensional, derivative-free optimization. This chapter offers a promising solution for tackling the challenges inherent in many-objective scenarios, potentially for overcoming the difficulties of applying the filter approach to problems with more than two objectives.

In Chapter 6, the integration of logarithmic barrier methods within the direct search framework is explored. This chapter provides valuable insights into how barrier techniques can be effectively combined with direct search strategies to improve performance in complex optimization contexts.

Finally, Chapter 7 presents concluding remarks and outlines some potential directions for future research. In addition, three appendices complement the main text: Appendices A and B provide further individual comparisons between solvers presented in Chapter 4 and Chapter 5, respectively, while Appendix C presents technical results and discusses the boundedness of Lagrange multipliers for the algorithm proposed in Section 6.1.

2

Optimality Conditions

In the domain of optimization, problems may or may not have constraints. In unconstrained optimization, where no constraints are imposed, the objective is to find the maximum or minimum of a function across the entire space of variables. Constrained optimization entails determining the point that satisfies the constraints and corresponds to the minimum or maximum value of the objective function. These constraints can be equalities, inequalities or a mix of them, defined by linear or nonlinear mathematical expressions. Constrained optimization problems are inherently more intricate than their unconstrained counterparts, due to the additional necessity of satisfying specific conditions. The inclusion of constraints introduces complexity, amplifying the difficulty of the problem from both theoretical and computational perspectives.

Constrained optimization is crucial when addressing real-world problems and plays a fundamental role across diverse domains. For instance, in engineering, design parameters are bound by physical limitations, and in finance, investment strategies must navigate budget constraints. Constrained optimization emerges as a valuable tool in incorporating these real-world limitations into the problem-solving process, ensuring practical and viable solutions.

In this chapter, our main focus is to present the necessary and sufficient conditions for a feasible point to be a local minimizer in the context of single-objective optimization and a Pareto point in a multiobjective setting. The chapter is divided into two main sections: one related to single-objective optimization (see Section 2.1) and another related to multiobjective optimization (see Section 2.2).

2.1 Single-objective optimization

Consider the general constrained, single-objective, nonlinear optimization problem, defined by:

$$\begin{aligned} &\min f(\boldsymbol{x}) \\ &\text{s. t. } \boldsymbol{x} \in \Omega \end{aligned} \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, and the set $\Omega \subseteq \mathbb{R}^n$ is the feasible set. The next definition provides a characterization of the solutions of Problem (2.1).

Definition 2.1.1. *Consider $f : \mathbb{R}^n \to \mathbb{R}$ and $\overline{x} \in \Omega \subseteq \mathbb{R}^n$. We say that $\overline{x} \in \Omega$ is a local minimizer of f in Ω when there is $\delta > 0$ such that $f(\overline{x}) \leq f(x)$ for all $x \in \Omega \cap B(\overline{x}, \delta)$. In the case that $f(\overline{x}) \leq f(x)$ for all $x \in \Omega$, we say that $\overline{x} \in \Omega$ is a global minimizer of f in Ω.*

It is worth mentioning that if the inequalities are strict in Definition 2.1.1 for $x \neq \overline{x}$, then the point $\overline{x}$ will be a strict minimizer.

Before delving into a detailed study of optimality conditions, we will first explore conditions that guarantee the existence of a solution for Problem (2.1). In other words, we will focus on conditions that enable us to ensure the existence of a solution for the nonlinear optimization Problem (2.1). Consequently, our analysis will involve studying the properties of the objective function and the characteristics of the imposed constraints.

The most well-known result regarding the existence of a solution is due to Weierstraß. The next theorem states this result and the proof can be found in [117, Thm. 2.3].

Theorem 2.1.2. *If the objective function f is lower semicontinuous in the feasible set Ω and Ω is a compact, non empty set, then the set of solutions of Problem (2.1) is non empty.*

Next, we define of a class of functions for which it is possible to guarantee the existence of a global minimizer, without the assumption of compactness of the feasible set.

Definition 2.1.3. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be coercive over Ω if for every sequence of feasible points $x_k \in \Omega$ such that $\lim_{k \to +\infty} \|x_k\| = +\infty$ we have $\lim_{k \to +\infty} f(x_k) = +\infty$, where $\|\cdot\|$ represents the Euclidean norm.*

The next theorem states the existence of a global minimizer, without assuming the compactness of the feasible region, by requiring coerciveness and lower semi-continuity of the objective function. This result can be found in [27, Prop. A.8].

Theorem 2.1.4. *If the objective function f is coercive and lower semicontinuous in the feasible set Ω and Ω is a closed, non empty set, then the set of optimum solutions of Problem (2.1) is a non empty set.*

One straightforward way of confirming the coerciveness of the objective function is to establish the boundedness of the level set $\mathcal{L}(c) = \{x \in \Omega \mid f(x) \leq c\}$, for any $c \in \mathbb{R}$.

Proposition 2.1.5. *[23, Prop. 11.12] A function $f : \mathbb{R}^n \to \mathbb{R}$ is coercive over Ω if, and only if, the level set $\mathcal{L}(c) = \{x \in \Omega \mid f(x) \leq c\}$ is bounded for any $c \in \mathbb{R}$.*

Under a convexity assumption, we can ensure that a local minimizer is also a global minimizer. Moreover, if the objective function is strictly convex, we also ensure the uniqueness of the solution.

Theorem 2.1.6. *[109, Thm. 2.1] Let Ω be a convex set and $\overline{\boldsymbol{x}}$ be a local minimizer of a convex function f. Then $\overline{\boldsymbol{x}}$ is also a global minimizer. If the f is strictly convex, then $\overline{\boldsymbol{x}}$ is the unique global minimizer over Ω.*

We will begin by examining optimality conditions for unconstrained optimization, as the optimality conditions for constrained problems are logic extensions of those for unconstrained problems. It is worth noting that Problem (2.1) is unconstrained when $\Omega = \mathbb{R}^n$.

The following definition characterizes directions, along which the objective function experiences a decrease, known as descent directions.

Definition 2.1.7. *A direction $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a descent direction for $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x} \in \mathbb{R}^n$, if there exists a $\delta > 0$ such that for all $\alpha \in]0, \delta]$,*

$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) < f(\boldsymbol{x}).$$

The set of all descent directions at $\boldsymbol{x} \in \mathbb{R}^n$ is a cone, denoted by $D(\boldsymbol{x})$. To recall, a nonempty subset $S \subset \mathbb{R}^n$ is a cone if, for all $t \geq 0$ and $\boldsymbol{y} \in S$, we have $t\boldsymbol{y} \in S$. The next theorem characterize descent directions for Problem (2.1), with $\Omega = \mathbb{R}^n$.

Theorem 2.1.8. *[24, Thm. 4.1.2] Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $\boldsymbol{x}$. If there is a vector $\boldsymbol{d} \in \mathbb{R}^n$ such that $\nabla f(\boldsymbol{x})^\top \boldsymbol{d} < 0$, then $\boldsymbol{d}$ is a descent direction for f at $\boldsymbol{x}$.*

The next result establishes that, at a local minimizer of Problem (2.1), each direction is non-descent, meaning that it is not possible to decrease the objective function value [109, p. 358].

Theorem 2.1.9. *If $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a local minimizer for Problem (2.1) with $\Omega = \mathbb{R}^n$, then $\nabla f(\overline{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0$, for all $\boldsymbol{d} \in \mathbb{R}^n$.*

Corollary 2.1.10 gives a first order necessary condition for $\overline{\boldsymbol{x}}$ to be a local unconstrained minimizer [24, Cor. of Thm. 4.1.2].

Corollary 2.1.10. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function at $\overline{\boldsymbol{x}}$. If $\overline{\boldsymbol{x}}$ is a local minimizer of f in $\mathbb{R}^n$, then*

$$\nabla f(\overline{\boldsymbol{x}}) = 0. \tag{2.2}$$

If condition (2.2) holds, then we say that $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a stationary, or a critical point, for Problem (2.1) with $\Omega = \mathbb{R}^n$. As previously mentioned, unless that f is convex, $\overline{\boldsymbol{x}}$ being a stationary point does not necessarily imply that $\overline{\boldsymbol{x}}$ is a minimizer. The next theorem states necessary and sufficient optimality conditions, when we assume the objective function f as being pseudoconvex[1]. For more details see [24, Thm. 4.1.5].

[1] Let S be a nonempty open set in $\mathbb{R}^n$, and let $f : S \to \mathbb{R}$ be differentiable on S. The function f is said to be pseudoconvex if for each $\boldsymbol{x}, \boldsymbol{y} \in S$ with $\nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \geq 0$, we have $f(\boldsymbol{y}) \geq f(\boldsymbol{x})$.

Theorem 2.1.11. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a pseudoconvex function. A point $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a global minimizer of f if and only if $\nabla f(\overline{\boldsymbol{x}}) = 0$.*

Let us now consider Problem (2.1), when the feasible region is defined by a general set of constraints, given by

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \text{ and } \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}\}$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^q$. Moreover, we assume that f, g_i $(i = 1, \ldots, p)$, and h_j $(j = 1, \ldots, q)$ are continuously differentiable functions. Denote by $I(\boldsymbol{x})$ the set of indices of active inequalities at a feasible point $\boldsymbol{x} \in \Omega$, i.e.,

$$I(\boldsymbol{x}) = \{i \in \{1, \ldots, p\} \mid g_i(\boldsymbol{x}) = 0\}.$$

All the subsequent results will be derived based on a cone approach. The first cone that we will consider is the tangent cone, defined by vectors that either tangentially touch or penetrate Ω. This cone serves as an approximation for the feasible set around a specific point, allowing sufficiently small steps within the feasible set without leaving it [152, Def. 12.2].

Definition 2.1.12 (Tangent cone)**.** *Let $\boldsymbol{x} \in \Omega$. The tangent cone of Ω around $\boldsymbol{x}$, denoted by $T_\Omega(\boldsymbol{x})$, is defined as:*

$$T_\Omega(\boldsymbol{x}) = \left\{\boldsymbol{d} \in \mathbb{R}^n \mid \exists\{\boldsymbol{x}_k\} \subset \Omega,\ \boldsymbol{x}_k \to \boldsymbol{x},\ \{t_k\} \subset \mathbb{R}_+,\ t_k \to 0,\ \text{such that } \frac{\boldsymbol{x}_k - \boldsymbol{x}}{t_k} \to \boldsymbol{d}\right\}$$

An alternative way to approximate the feasible set Ω involves computing a linearizing cone. This cone serves as a linear approximation to the feasible region defined by Ω, where active inequality constraints and all equality constraints are linearized [152, Def. 12.3].

Definition 2.1.13 (Linearizing cone)**.** *Let $\boldsymbol{x} \in \Omega$. The linearizing cone of Ω computed at $\boldsymbol{x}$, denoted by $L_\Omega(\boldsymbol{x})$, is defined as:*

$$L_\Omega(\boldsymbol{x}) = \{\boldsymbol{d} \in \mathbb{R}^n \mid \nabla g_i(\boldsymbol{x})^\top \boldsymbol{d} \leq 0, \forall i \in I(\boldsymbol{x}) \text{ and } \nabla h_j(\boldsymbol{x})^\top \boldsymbol{d} = 0,\ \forall j = 1, \ldots, q\}$$

The following lemma is a well-known result in the specialized literature about the relationship between the two cones $T_\Omega(\boldsymbol{x})$ and $L_\Omega(\boldsymbol{x})$ defined before. This result can be found in [152, Lem. 12.2 (i)].

Lemma 2.1.14. *Let $\boldsymbol{x} \in \Omega$, then $T_\Omega(\boldsymbol{x}) \subseteq L_\Omega(\boldsymbol{x})$.*

The next result establishes that in a local minimizer of Problem (2.1), each tangent direction is non-descent, meaning that it is not possible to simultaneously keep feasibility and decrease in the objective function value. This result can also be regarded as a necessary optimality condition [152, Thm. 12.3].

Theorem 2.1.15. *If $\overline{\boldsymbol{x}} \in \Omega$ is a local minimizer for Problem (2.1), then $\nabla f(\overline{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0$, for all $\boldsymbol{d} \in T_\Omega(\overline{\boldsymbol{x}})$.*

To derive the Fritz-John necessary optimality conditions, we present the following auxiliary result.

Lemma 2.1.16. *If $\overline{x} \in \Omega$ is a local minimizer for Problem (2.1), then the system*

$$\begin{aligned} &\nabla f(\overline{x})^\top \boldsymbol{d} < 0 \\ &\nabla g_i(\overline{x})^\top \boldsymbol{d} \leq 0,\ i \in I(\overline{x}) \\ &\nabla h_j(\overline{x})^\top \boldsymbol{d} = 0,\ j = 1, \ldots, q \end{aligned} \tag{2.3}$$

has no solution $\boldsymbol{d} \in \mathbb{R}^n$.

Now, by applying Motzkin's Theorem [141, Thm. 2] to system (2.3), we obtain the Fritz-John necessary optimality conditions [24, Thm. 4.3.2].

Theorem 2.1.17 (Fritz-John necessary optimality conditions)**.** *Let $\overline{x} \in \Omega$. If $\overline{x}$ is a local minimizer for Problem (2.1), then there exists $\theta \in \mathbb{R}_+$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, and $\boldsymbol{\mu} \in \mathbb{R}^q$ not all nulls such that*

$$\begin{aligned} &\theta \nabla f(\overline{x}) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\overline{x}) = \mathbf{0} \\ &\lambda_i g_i(\overline{x}) = 0,\ i = 1, \ldots, p. \end{aligned} \tag{2.4}$$

Note that if the multiplier associated with the objective function, denoted as θ, is zero, then all information pertaining to the objective function is lost. If a constraint qualification (CQ) is added to the hypotheses of the Fritz-John theorem, we have the insurance of the multiplier θ being non-zero.

Let us consider the most general constraint qualification (CQ) [104], which encompasses several other constraint qualifications as special cases.

Definition 2.1.18 (Guignard Constraint Qualification - GCQ)**.**

$$(T_\Omega(\overline{x}))^\circ = (L(\overline{x}))^\circ,$$

where $(\cdot)^\circ$ denotes the polar cone[2]*.*

Under GCQ, we can now ensure that the multiplier θ, associated with the objective function is non-null, a result formalized in Theorem 2.1.19 (see [19, Thm. 3.8]).

Theorem 2.1.19 (Karush-Kuhn-Tucker necessary optimality conditions (KKT))**.** *Let $\overline{x} \in \Omega$ be a local minimizer for Problem (2.1), and suppose that the (GCQ) holds. Then, there exists $\boldsymbol{\lambda} \in \mathbb{R}^p_+$ and $\boldsymbol{\mu} \in \mathbb{R}^q$ such that*

$$\begin{aligned} &\nabla f(\overline{x}) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\overline{x}) = \mathbf{0} \\ &\lambda_i g_i(\overline{x}) = 0,\ i = 1, \ldots, p. \end{aligned} \tag{2.5}$$

[2]Let $S \subset \mathbb{R}^n$, the polar cone of S is defined by $(S)^\circ = \{\, \mathbf{p} \in \mathbb{R}^n \mid \mathbf{p}^\top \boldsymbol{x} \leq 0, \forall \boldsymbol{x} \in S \}$.

Feasible points that satisfy (2.5) under a (CQ) are referred to as stationary points or KKT points for Problem (2.1).

In Theorem 2.1.19, the KKT conditions are established as necessary for local optimality under a specific constraint qualification (GCQ). Although the (GCQ) condition is general, it is difficult to verify in practice, since it requires a detailed analysis of the tangent cone and the associated polar cone. Hence, alternative constraint qualification conditions, such as the linear independence constraint qualification (LICQ), the Mangasarian-Fromovitz constraint qualification (MFCQ), the constant rank constraint qualification (CRCQ), etc., can be considered [24]. These conditions, while less general, are more straightforward to verify.

Now, we will see sufficient optimality conditions that enable us to guarantee that a feasible point is a local minimizer. For more details (see [24, Thm. 4.3.6]).

Theorem 2.1.20 (Fritz-John sufficient optimality conditions)**.** *Assume that $\overline{\boldsymbol{x}} \in \Omega$ satisfies (2.4). If h_j, $j = 1 \ldots, q$ are affine and $\nabla h_j(\overline{\boldsymbol{x}})$ are linear independent, and if there exists a neighborhood $B(\overline{\boldsymbol{x}}, \varepsilon)$ of $\overline{\boldsymbol{x}} \in \Omega$ such that f is pseudoconvex on $\Omega \cap B(\overline{\boldsymbol{x}}, \varepsilon)$, and g_i, for $i \in I(\overline{\boldsymbol{x}})$, are strictly pseudoconvex over $\Omega \cap B(\overline{\boldsymbol{x}}, \varepsilon)$, then $\overline{\boldsymbol{x}} \in \Omega$ is a local minimizer for Problem (2.1).*

The following theorem states that given relatively mild convexity assumptions on f, g_i $(i = 1, \ldots, p)$, and h_j $(j = 1, \ldots, q)$, the KKT conditions are not only necessary but also sufficient for local optimality, see [24, Thm. 4.3.8].

Theorem 2.1.21 (Karush-Kuhn-Tucker sufficient optimality conditions (KKT))**.** *Let $\overline{\boldsymbol{x}} \in \Omega$. Suppose that $\overline{\boldsymbol{x}}$ satisfies (2.5). Define $J = \{j \mid \lambda_j \neq 0\}$. Assume that f is pseudoconvex, g_i is quasiconvex[3] for $i \in I(\overline{\boldsymbol{x}})$, and h_j is quasiconcave for $j \in J$. Then, $\overline{\boldsymbol{x}}$ is a global optimal solution of Problem (2.1). In particular, if the generalized convexity assumptions on the objective and constraints functions are restricted to the domain $B(\overline{\boldsymbol{x}}, \varepsilon)$, then $\overline{\boldsymbol{x}}$ is a local minimizer for Problem (2.1).*

In the next definition, we present the Mangasarian-Fromovitz constraint qualification (MFCQ), for a problem that has an additional convex set of constraints, i.e., the feasible region of Problem (2.1) is given by $\Omega \cap X$ where X is a closed convex set.

Definition 2.1.22. *The point $\boldsymbol{x} \in X$ is said to satisfy the MFCQ for Problem (2.1) if the two following conditions are satisfied:*

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

$$\left(\sum_{j=1}^{q} \alpha_j \nabla h_j(\boldsymbol{x})\right)^{\top} \boldsymbol{d} \geq 0, \qquad \forall \boldsymbol{d} \in T_X(\boldsymbol{x}); \tag{2.6}$$

[3] Let $f : S \to \mathbb{R}$, where S is a nonempty convex set in $\mathbb{R}^n$. The function f is said to be quasiconvex if for each $\boldsymbol{x}, \boldsymbol{y} \in S$, the inequality $f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \max\{f(\boldsymbol{x}), f(\boldsymbol{y})\}$ holds for each $\lambda \in (0, 1)$. The function f is said to be quasiconcave if $-f$ is quasiconvex.

(b) There exists a feasible direction $\boldsymbol{d} \in T_X(\boldsymbol{x})$*, such that:*

$$\nabla g_i(\boldsymbol{x})^\top \boldsymbol{d} < 0, \quad \forall i \in I_+(\boldsymbol{x}), \quad \nabla h_j(\boldsymbol{x})^\top \boldsymbol{d} = 0, \quad \forall j = 1, \ldots, q \tag{2.7}$$

where $I_+(\boldsymbol{x}) = \{i \mid g_i(\boldsymbol{x}) \geq 0\}$.

Consider the Lagrangian function $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, associated with the nonlinear constraints of Problem (2.1) with the feasible region replaced by $\Omega \cap X$, defined by:

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

The following proposition is a well-known result (see [27, Sec. 3.3]), which states necessary optimality conditions for Problem (2.1) with feasible region defined as $\Omega \cap X$.

Proposition 2.1.23. *Let* $\overline{\boldsymbol{x}} \in \Omega \cap X$ *be a local minimizer of Problem* (2.1) *that satisfies the MFCQ. Then, there exist vectors* $\overline{\boldsymbol{\lambda}} \in \mathbb{R}^p$, $\overline{\boldsymbol{\mu}} \in \mathbb{R}^q$ *such that*

$$\nabla_x L(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}}, \overline{\boldsymbol{\mu}})^\top (\boldsymbol{x} - \overline{\boldsymbol{x}}) \geq 0, \qquad \forall \boldsymbol{x} \in X \tag{2.8}$$

$$\overline{\lambda}_i g_i(\overline{\boldsymbol{x}}) = 0, \quad \overline{\lambda}_i \geq 0 \text{ for } i = 1, \ldots, p. \tag{2.9}$$

Therefore, we consider the following definition of stationarity.

Definition 2.1.24 (Stationary point)**.** *A feasible point* $\overline{\boldsymbol{x}} \in \Omega \cap X$ *is said to be a stationary point for Problem* (2.1) *if vectors* $\overline{\boldsymbol{\lambda}} \in \mathbb{R}^p$ *and* $\overline{\boldsymbol{\mu}} \in \mathbb{R}^q$ *exist such that* (2.8) *and* (2.9) *are satisfied.*

All the results stated so far are based on the assumption that the functions involved are continuously differentiable. However, there may be situations where this assumption does not hold. Following [14, 20, 47, 60, 115], we will now state conditions to determine whether a point is an optimal solution when the smoothness assumption is not present.

Let us begin by revisiting an unconstrained problem, i.e., Problem (2.1) with $\Omega = \mathbb{R}^n$. In order to derive optimality conditions for any locally Lipschitz continuous function, the concept of directional derivative is essential. The next definition is a generalization of the classical definition of directional derivative, known as the Clarke generalized directional derivative [14, Def. 6.4].

Definition 2.1.25 (Generalized Directional Derivative)**.** *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be Lipschitz continuous near* $\boldsymbol{x} \in \mathbb{R}^n$*. The generalized directional derivative of* f *at* $\boldsymbol{x}$ *in the direction* $\boldsymbol{d} \in \mathbb{R}^n$ *is:*

$$f^\circ(\boldsymbol{x}; \boldsymbol{d}) = \limsup_{\boldsymbol{y} \to \boldsymbol{x}, t \downarrow 0} \frac{f(\boldsymbol{y} + t\boldsymbol{d}) - f(\boldsymbol{y})}{t}. \tag{2.10}$$

The next definition formalizes the concept of Clarke subdifferential, also known as the generalized gradient, and will be particularly useful for obtaining optimality conditions.

Definition 2.1.26 (Clarke Subdifferential)**.** *For any* $\boldsymbol{x} \in \mathbb{R}^n$ *the Clarke subdifferential is defined as the set:*

$$\partial f(\boldsymbol{x}) := \{\boldsymbol{v} \in \mathbb{R}^n \mid f^\circ(\boldsymbol{x}; \boldsymbol{d}) \geq \boldsymbol{d}^\top \boldsymbol{v}, \text{ for all } \boldsymbol{d} \in \mathbb{R}^n\}.$$

When f is a strictly differentiable function at the point $\boldsymbol{x}$, the subdifferential of f at point $\boldsymbol{x}$ reduces to a singleton.

Proposition 2.1.27. *[47, Prop. 2.2.4] If f is strictly differentiable at $\boldsymbol{x}$, then f is Lipschitz continuous near $\boldsymbol{x}$ and $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$. Conversely, if f is Lipschitz near $\boldsymbol{x}$ and $\partial f(\boldsymbol{x})$ reduces to a singleton, then f is strictly differentiable at $\boldsymbol{x}$ and $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$.*

Now, we have all the ingredients to establish some optimality conditions. The next theorem establishes a necessary condition for a locally Lipschitz continuous function to attain a local minimum.

Theorem 2.1.28. *[20, Thm. 4.1] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\overline{\boldsymbol{x}} \in \mathbb{R}^n$. If f attains a local minimum at $\overline{\boldsymbol{x}}$, then*

$$\mathbf{0} \in \partial f(\overline{\boldsymbol{x}}) \text{ and } f^\circ(\overline{\boldsymbol{x}}; \boldsymbol{d}) \geq 0 \text{ for all } \boldsymbol{d} \in \mathbb{R}^n. \tag{2.11}$$

A point $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ satisfying the property $\mathbf{0} \in \partial f(\overline{\boldsymbol{x}})$ is called a stationary point of f. In the next theorem we have a characterization of descent directions.

Theorem 2.1.29. *[20, Thm. 4.5] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\boldsymbol{x} \in \mathbb{R}^n$. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a descent direction for f at $\boldsymbol{x}$ if*

$$\boldsymbol{v}^\top \boldsymbol{d} < 0 \text{ for all } \boldsymbol{v} \in \partial f(\boldsymbol{x}) \text{ or } f^\circ(\boldsymbol{x}; \boldsymbol{d}) < 0.$$

The next corollary states that either we encounter a stationary point, indicating a potential solution, or we can find a descent direction.

Corollary 2.1.30. *[20, Cor. 4.1] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\boldsymbol{x} \in \mathbb{R}^n$. Then either $\mathbf{0} \in \partial f(\boldsymbol{x})$ or there is a descent direction $\boldsymbol{d} \in \mathbb{R}^n$ for f at $\boldsymbol{x}$.*

Considering Problem (2.1) and Definition 2.1.7, the following definition introduces the polar cone of subdifferential directions at a given point.

Definition 2.1.31. *The polar cone of subdifferential directions at $\boldsymbol{x} \in \mathbb{R}^n$ is*

$$D^\circ(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^n \mid \boldsymbol{d} = \mathbf{0} \text{ or } \boldsymbol{v}^\top \boldsymbol{d} < 0 \text{ for all } \boldsymbol{v} \in \partial f(\boldsymbol{x})\}.$$

From Theorem 2.1.29, it is easy to see that the following inclusion holds $D^\circ(\boldsymbol{x}) \subseteq D(\boldsymbol{x})$. In the next corollary, we state a geometrical optimality condition.

Corollary 2.1.32. *[20, Cor. 4.3] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function at $\overline{\boldsymbol{x}} \in \mathbb{R}^n$. If f attains a local minimum at $\overline{\boldsymbol{x}}$, then*

$$D^\circ(\overline{\boldsymbol{x}}) = D(\overline{\boldsymbol{x}}) = \{\mathbf{0}\}. \tag{2.12}$$

If, in addition, f is convex, then condition (2.12) is sufficient for $\overline{\boldsymbol{x}}$ to be a global minimizer of f.

If $\Omega \neq \mathbb{R}^n$, then the descent directions must be also feasible, at least locally. In the next definition we formalize the concept of cone of locally feasible directions, $\mathcal{F}_\Omega$.

Definition 2.1.33. *The cone of locally feasible directions of the nonempty set Ω at $\boldsymbol{x} \in \Omega$ is*

$$\mathcal{F}_\Omega(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^n \mid \text{there exist } \delta > 0 \text{ such that } \boldsymbol{x} + t\boldsymbol{d} \in \Omega \text{ for all } t \in]0, \delta]\}. \tag{2.13}$$

Taking into account this new definition, it is possible to extend the geometrical optimality conditions to the general constrained case, i.e., when $\Omega \subset \mathbb{R}^n$.

Corollary 2.1.34. *[20, Thm. 4.6] Let $\overline{\boldsymbol{x}} \in \Omega \neq \emptyset$ be a local minimizer for Problem (2.1), where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function at $\overline{\boldsymbol{x}}$. Then*

$$D^\circ(\overline{\boldsymbol{x}}) \cap \mathcal{F}_\Omega(\overline{\boldsymbol{x}}) = \{\boldsymbol{0}\}. \tag{2.14}$$

If, in addition, Problem (2.1) is convex, then condition (2.14) implies that $\overline{\boldsymbol{x}} \in \Omega$ is a global minimizer for Problem (2.1).

To extend the optimality conditions given in Theorem 2.1.15 to a nonsmooth setting, we will need a generalization of the tangent cone commonly used in nonlinear programming, namely the Clarke's tangent cone [46].

Definition 2.1.35. *A vector $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subset \mathbb{R}^n$ at the point $\boldsymbol{x} \in \mathrm{cl}(\Omega)$ if for every sequence $\{\boldsymbol{y}_k\}$ of elements of Ω that converges to $\boldsymbol{x}$ and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{\boldsymbol{w}_k\}$ converging to $\boldsymbol{d}$ such that $\boldsymbol{y}_k + t_k\boldsymbol{w}_k \in \Omega$.*

The set of all Clarke tangent vectors to Ω at $\boldsymbol{x}$ is called the Clarke tangent cone to Ω at $\boldsymbol{x}$ and is denoted by $T_\Omega^{Cl}(\boldsymbol{x})$. The tangent cone is the closure of another relevant cone for the following analysis, namely the Clarke's hypertangent cone [47].

Definition 2.1.36. *A vector $\boldsymbol{d} \in \mathbb{R}^n$ is said to be a Clarke hypertangent vector to the set $\Omega \subset \mathbb{R}^n$ at the point $\boldsymbol{x} \in \Omega$ if there exists a scalar $\epsilon > 0$ such that $\boldsymbol{y} + t\boldsymbol{w} \in \Omega$ for all $\boldsymbol{y} \in \Omega \cap B(\boldsymbol{x}; \epsilon)$, $\boldsymbol{w} \in B(\boldsymbol{d}; \epsilon)$, and $0 < t < \epsilon$.*

The set of all hypertangent vectors to Ω at $\boldsymbol{x}$ is called the hypertangent cone to Ω at $\boldsymbol{x}$ and is denoted by $H_\Omega^{Cl}(\boldsymbol{x})$. Whenever the interior of $T_\Omega^{Cl}(\boldsymbol{x})$ is nonempty, $H_\Omega^{Cl}(\boldsymbol{x}) = \mathrm{int}(T_\Omega^{Cl}(\boldsymbol{x}))$.

The notion of generalized directional derivative also needs to be adapted to account for the presence of constraints [47, 117].

Definition 2.1.37. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz continuous near $\boldsymbol{x} \in \Omega$. The Clarke-Jahn generalized directional derivative of f along $\boldsymbol{d} \in H_\Omega^{Cl}(\boldsymbol{x})$ is defined as:*

$$f^\circ(\boldsymbol{x}; \boldsymbol{d}) := \limsup_{\substack{\boldsymbol{y} \to \boldsymbol{x}, \boldsymbol{y} \in \Omega \\ t \downarrow 0, \boldsymbol{y} + t\boldsymbol{d} \in \Omega}} \frac{f(\boldsymbol{y} + t\boldsymbol{d}) - f(\boldsymbol{y})}{t}.$$

When $H_\Omega^{Cl}(\boldsymbol{x})$ is nonempty, the Clarke-Jahn generalized directional derivatives along directions $\boldsymbol{v}$ in $T_\Omega^{Cl}(\boldsymbol{x})$ can be computed by taking limits of sequences of directions belonging to the hypertangent cone [13].

Proposition 2.1.38. *[13, Thm. 3.9] Let $f : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz continuous near $\boldsymbol{x} \in \Omega$ and assume that $H_\Omega^{Cl}(\boldsymbol{x})$ is nonempty. The Clarke-Jahn generalized directional derivative of f along $\boldsymbol{v} \in T_\Omega^{Cl}(\boldsymbol{x})$ can be computed as:*

$$f^\circ(\boldsymbol{x}; \boldsymbol{v}) := \lim_{\substack{\boldsymbol{d} \to \boldsymbol{v} \\ \boldsymbol{d} \in H_\Omega^{Cl}(\boldsymbol{x})}} f^\circ(\boldsymbol{x}; \boldsymbol{d}).$$

In the next theorem, we present a necessary optimality condition that is satisfied when $\overline{\boldsymbol{x}} \in \Omega$ is a local minimizer. For more details see [13, eq. 1.4].

Theorem 2.1.39. *If f is Lipschitz continuous near $\overline{\boldsymbol{x}} \in \Omega$, a local minimizer for Problem (2.1), then $f^\circ(\overline{\boldsymbol{x}}; \boldsymbol{d}) \geq 0$ for every direction $\boldsymbol{d} \in T_\Omega^{Cl}(\overline{\boldsymbol{x}})$.*

Now, consider that the feasible region of Problem (2.1) is given by

$$\Omega = \{\boldsymbol{x} \in X \mid \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}\}, \tag{2.15}$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ and $X \subseteq \mathbb{R}^n$ is a nonempty set. Moreover, assume that f, g_i $(i = 1, \ldots, p)$ are locally Lipschitz continuous.

Definition 2.1.40. *The polar cone of constraint subgradient directions at $\boldsymbol{x} \in \Omega$ is*

$$D_\Omega^\circ(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^n \mid \boldsymbol{d} = \boldsymbol{0} \text{ or } \boldsymbol{v}_i^\top \boldsymbol{d} < 0 \text{ for all } \boldsymbol{v}_i \in \partial g_i(\boldsymbol{x}) \text{ and } i \in I(\boldsymbol{x})\},$$

where $I(\boldsymbol{x}) = \{i \in \{1, \ldots, p\} \mid g_i(\boldsymbol{x}) = 0\}$.

The next theorem is an extension of the geometrical optimality conditions for the constrained case, when Ω is defined as (2.15).

Theorem 2.1.41. *[20, Thm. 4.9] Let $\overline{\boldsymbol{x}} \in \Omega$ be a local minimizer for Problem (2.1), where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, p$ are locally Lipschitz continuous functions at $\overline{\boldsymbol{x}} \in \Omega$. Then*

$$D_\Omega^\circ(\overline{\boldsymbol{x}}) \cap D(\overline{\boldsymbol{x}}) = \{\boldsymbol{0}\}. \tag{2.16}$$

However, even in the convex case, the condition (2.16) is not sufficient for optimality.

In the next theorem, we have the Fritz-John necessary optimality conditions in the nonsmooth case [20, Thm. 4.10]

Theorem 2.1.42 (Fritz-John optimality conditions)**.** *Let $\overline{\boldsymbol{x}} \in \Omega$ be a local minimizer of Problem (2.1), where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, p$ are locally Lipschitz continuous functions at $\overline{\boldsymbol{x}}$. Then, there exist multipliers $\theta, \lambda_i \geq 0$ for $i = 1, \ldots, p$ such that $\theta + \sum_{i=1}^p \lambda_i = 1$, $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ for $i = 1, \ldots, p$ and*

$$\boldsymbol{0} \in \theta \partial f(\overline{\boldsymbol{x}}) + \sum_{i=1}^p \lambda_i \partial g_i(\overline{\boldsymbol{x}}). \tag{2.17}$$

As mentioned for smooth optimization, the Fritz-John optimality conditions do not guarantee that the Lagrange multiplier associated with the objective function is not null, resulting in a loss of information about the objective function. To address this issue, a constraint qualification is required.

In the following two definitions, we state two distinct constraint qualifications. The first is the Slater constraint qualification, which requires guaranteeing the existence of a point in the feasible region that satisfies the inequality constraints strictly.

Definition 2.1.43. *[20, Def. 4.8] Problem (2.1) satisfies the Slater constraint qualification if there exists $\boldsymbol{x} \in \Omega$ such that $\boldsymbol{g}(\boldsymbol{x}) < \mathbf{0}$.*

The next definition states the Cottle constraint qualification, which is defined pointwise.

Definition 2.1.44. *[20, Def. 4.9] Problem (2.1) satisfies the Cottle constraint qualification at $\boldsymbol{x} \in \Omega$ if either $\boldsymbol{g}(\boldsymbol{x}) < \mathbf{0}$ or $\mathbf{0} \notin \partial g_i(\boldsymbol{x})$ for $i = 1, \ldots, p$.*

Clearly, if Problem (2.1) satisfies the Cottle constraint qualification at some point $\boldsymbol{x} \in \Omega$, then it also satisfies the Slater constraint qualification. Furthermore, if functions g_i are convex for all $i = 1, \ldots, p$ and the problem satisfies the Slater constraint qualification, then it also satisfies the Cottle constraint qualification at every $\boldsymbol{x} \in \Omega$, [20, Lem. 4.3].

By assuming the Cottle constraint qualification, we can guarantee that the multiplier θ associated with the objective function is non null, as formalized in Theorem 2.1.45 (see [20, Thm. 4.11]).

Theorem 2.1.45 (KKT Necessary Optimality Conditions)**.** *Let Problem (2.1) satisfy the Cottle constraint qualification at a local minimizer $\overline{\boldsymbol{x}} \in \Omega$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, p$ are locally Lipschitz continuous functions. Then there exist multipliers $\lambda_i \geq 0$, for $i = 1, \ldots, p$ such that $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ and*

$$\mathbf{0} \in \partial f(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{\boldsymbol{x}}).$$

Under the convexity assumption we can establish sufficient optimality conditions [20, Thm. 4.12].

Theorem 2.1.46 (KKT Sufficient Optimality Conditions)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, p$ be convex functions. If at $\overline{\boldsymbol{x}} \in \Omega$ there exist multipliers $\lambda_i \geq 0$, for $i = 1, \ldots, p$ such that $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ and*

$$\mathbf{0} \in \partial f(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{\boldsymbol{x}}),$$

then $\overline{\boldsymbol{x}} \in \Omega$ is a global minimizer of Problem (2.1).

Combining Theorems 2.1.45 and 2.1.46 we have necessary and sufficient optimality conditions for $\overline{\boldsymbol{x}} \in \Omega$ be a minimizer of Problem (2.1).

The next proposition can be regarded as a Clarke Fritz-John optimality condition, since it is not possible to ensure that the Lagrange multiplier associated with the objective function is different than zero. More details can be found in [20].

Proposition 2.1.47. *[83, Fritz-John optimality conditions, Prop. 3.1] Let $\overline{x} \in \Omega$ be a local minimizer of Problem (2.1). Then, multipliers θ, and λ_i for $i = 1, \ldots, p$ not all zero exist, with*

$$\theta \geq 0, \quad \lambda_i \geq 0, \text{ and } \lambda_i g_i(\overline{x}) = 0 \text{ for } i = 1, \ldots, p$$

such that for every $\boldsymbol{d} \in \mathcal{F}_X(\overline{x})$,

$$\max \left\{ \boldsymbol{v}^\top \boldsymbol{d} \mid \boldsymbol{v} \in \theta \partial f(\overline{x}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{x}) \right\} \geq 0.$$

The next definition is an extension of the Mangasarian-Fromovitz constraints qualification MFCQ for nonsmooth problems.

Definition 2.1.48 (Extended Mangasarian-Fromovitz constraint qualification (EMFCQ))**.** *Given Problem (2.1), for any $\boldsymbol{x} \in \Omega$ a direction $\boldsymbol{d} \in \mathcal{F}_X(\boldsymbol{x})$ exists such that for all $i \in I(\boldsymbol{x})$:*

$$\boldsymbol{v}^\top \boldsymbol{d} < 0 \quad \forall \boldsymbol{v} \in \partial g_i(\boldsymbol{x}). \tag{2.18}$$

Adding the hypothesis of EMFCQ, it is possible to ensure that the Lagrange multiplier associated with the objective function is nonzero, establishing the Clarke-KKT necessary optimality conditions (see [83, Cor. 3.2]).

Proposition 2.1.49 (Clarke-KKT necessary optimality conditions)**.** *Let $\overline{x} \in \Omega$ be a local minimizer of Problem (2.1) and assume that $\overline{x}$ satisfies EMFCQ. Then, multipliers $\lambda_i \in \mathbb{R}$ for $i = 1, \ldots, p$ exist, with*

$$\lambda_i \geq 0 \text{ and } \lambda_i g_i(\overline{x}) = 0, \text{ for } i = 1, \ldots, p,$$

such that for every $\boldsymbol{d} \in \mathcal{F}_X(\overline{x})$

$$\max \left\{ \boldsymbol{v}^\top \boldsymbol{d} \mid \boldsymbol{v} \in \partial f(\overline{x}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{x}) \right\} \geq 0. \tag{2.19}$$

The next definition characterizes a stationary point and can be found in [83].

Definition 2.1.50 (Stationary point)**.** *Given Problem (2.1), the feasible point $\overline{x}$ is said to be a stationary point if multipliers $\lambda_1, \lambda_2, \ldots, \lambda_p \in \mathbb{R}$ exist, with*

$$\lambda_i \geq 0 \text{ and } \lambda_i g_i(\overline{x}) = 0 \text{ for } i = 1, \ldots, p$$

such that for every $\boldsymbol{d} \in \mathcal{F}_X(\overline{x})$

$$\max \left\{ \boldsymbol{v}^\top \boldsymbol{d} \mid \boldsymbol{v} \in \partial f(\overline{x}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{x}) \right\} \geq 0. \tag{2.20}$$

2.2 Multiobjective optimization

In the previous subsection, we discussed necessary and sufficient optimality conditions for single-objective optimization problems, addressing unconstrained and constrained optimization, both for the smooth and nonsmooth cases. The present subsection is dedicated to optimality conditions for multiobjective optimization problems.

As in Section 2.1, we will start by the definition of solution. In order to do that, consider the general multiobjective optimization problem given by

$$\begin{aligned} &\min \ \mathbf{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^\top \\ &\text{s. t. } \boldsymbol{x} \in \Omega, \end{aligned} \tag{2.21}$$

where $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$, with $m \in \mathbb{N}$, $m \geq 2$, and $\Omega \subseteq \mathbb{R}^n$ is the feasible set.

In multiobjective optimization, the concept of Pareto dominance is essential for point comparison. To describe it, we will make use of the strict partial order induced by the cone $\mathbb{R}^m_+ = \{\boldsymbol{z} \in \mathbb{R}^m \mid \boldsymbol{z} \geq 0\}$, defined as:

$$\mathbf{F}(\boldsymbol{x}) \preceq_{\mathbf{F}} \mathbf{F}(\boldsymbol{y}) \Leftrightarrow \mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) \in \operatorname{int}(\mathbb{R}^m_+), \tag{2.22}$$

$$\mathbf{F}(\boldsymbol{x}) \prec_{\mathbf{F}} \mathbf{F}(\boldsymbol{y}) \Leftrightarrow \mathbf{F}(\boldsymbol{y}) - \mathbf{F}(\boldsymbol{x}) \in \mathbb{R}^m_+ \backslash \{\mathbf{0}\}. \tag{2.23}$$

Given two points $\boldsymbol{x}, \boldsymbol{y}$ in Ω, we say that $\boldsymbol{x}$ dominates $\boldsymbol{y}$, i.e. $\boldsymbol{x} \prec_{\mathbf{F}} \boldsymbol{y}$, when $\mathbf{F}(\boldsymbol{x}) \prec_{\mathbf{F}} \mathbf{F}(\boldsymbol{y})$. Moreover, we say that $\boldsymbol{x}$ weakly dominates $\boldsymbol{y}$, i.e. $\boldsymbol{x} \preceq_{\mathbf{F}} \boldsymbol{y}$, when $\mathbf{F}(\boldsymbol{x}) \preceq_{\mathbf{F}} \mathbf{F}(\boldsymbol{y})$.

We are now in conditions of characterizing (weak) efficient points of Problem (2.21).

Definition 2.2.1 (Efficient/Pareto point)**.** *A point $\overline{\boldsymbol{x}} \in \Omega$ is said to be a global (weakly) efficient point of Problem (2.21) if there is no $\boldsymbol{y} \in \Omega$ such that ($\boldsymbol{y} \preceq_{\boldsymbol{F}} \overline{\boldsymbol{x}}$) $\boldsymbol{y} \prec_{\boldsymbol{F}} \overline{\boldsymbol{x}}$. If there exists a neighborhood $N(\overline{\boldsymbol{x}})$ of $\overline{\boldsymbol{x}}$ such that the previous property holds in $\Omega \cap N(\overline{\boldsymbol{x}})$, then $\overline{\boldsymbol{x}}$ is called a local (weak) efficient point of Problem (2.21).*

In other words, a solution to Problem (2.21) is called efficient when it is impossible to improve any component of objective function without deteriorating another. The image of the set of global efficient points for Problem (2.21) constitutes the solution of the multiobjective optimization problem and is denoted by the Pareto front.

It is easy to prove that an efficient optimal solution is also weak efficient optimal. However, the following well know example shows that the counterpart is not always true.

Example 2.2.2 (Adapted from Moulin and Soulié [149])**.** *Consider the following problem*

$$\begin{aligned} &\min \ \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))^\top \\ &\text{s. t. } x \in \Omega, \end{aligned} \tag{2.24}$$

where $\boldsymbol{F} : \mathbb{R}^2 \to \mathbb{R}^2$ with $f_1(\boldsymbol{x}) = x_1$ and $f_2(\boldsymbol{x}) = x_2$. Moreover, the feasible set is given by $\Omega = \{\boldsymbol{x} \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\} \cup [0, 1] \times [-1, 0] \subseteq \mathbb{R}^2$.

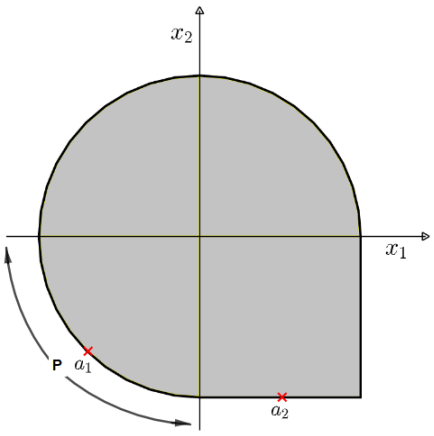


Figure 2.1: Feasible set Ω.

In Figure 2.1, the point $\mathbf{a}_1 = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$ can not strictly improve f_1 without strictly worsening f_2. The point $\mathbf{a}_1$ is an efficient solution. The set of efficient solutions, P, is the arc between $(-1, 0)$ and $(0, -1)$. Note that $\mathbf{a}_2 = \left(\frac{1}{2}, -1\right)$ is a weakly efficient solution of Problem (2.24), and that the set P_f defined by $P_f = P \cup \{(\lambda, -1) : 0 \leq \lambda \leq 1\}$ is the set of weakly efficient solutions of Problem (2.24). Point $\mathbf{a}_2 \in P_f$ but $\mathbf{a}_2 \notin P$.

The next proposition presents a well-known result regarding the existence of efficient solutions. A similar result, although based on a weaker assumption concerning the objective function can be found in [185, Thm. 7].

Proposition 2.2.3. *Let $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m$ be a continuous function and the feasible region Ω a nonempty compact set. Then Problem (2.21) has efficient (Pareto) solutions.*

In Theorem 2.1.6, we saw that convexity assumptions allow to ensure that a local minimizer is also a global minimizer for a single-objective optimization problem. The next theorem states the counterpart for multiobjective optimization.

Theorem 2.2.4. *[144, Thm. 2.2.3] Let $\overline{\boldsymbol{x}}$ be a local efficient solution of a convex multiobjective optimization problem. Then $\overline{\boldsymbol{x}}$ is also a global efficient solution.*

The aforementioned result can be derived under slightly weaker assumptions. For example, assuming the convexity of the feasible region and the quasiconvexity of the objective function components, with the added condition that at least one component is strictly quasiconvex. For more details, see [144, Thm. 2.2.4].

As in Section 2.1, we will start by establishing optimality conditions for Problem (2.21) with $\Omega = \mathbb{R}^n$. The definition of a descent direction, as given in Definition 2.1.7 for the single-objective case, can be restated as follows.

Definition 2.2.5. *A direction $\boldsymbol{d} \in \mathbb{R}^n$ is called a descent direction for $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^m$ at $\boldsymbol{x} \in \mathbb{R}^n$, if there exists a $\delta > 0$ such that for all $\alpha \in]0, \delta]$,*

$$\boldsymbol{F}(\boldsymbol{x} + \alpha \boldsymbol{d}) < \boldsymbol{F}(\boldsymbol{x}),$$

where the inequality should be regarded component-wise.

The following theorem sets a condition, both sufficient and necessary, for a direction to qualify as a descent direction [90].

Theorem 2.2.6. *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions for $\ell = 1, \ldots, m$. The direction $\boldsymbol{d} \in \mathbb{R}^n$ is a descent direction for the function $\boldsymbol{F}$ at $\boldsymbol{x} \in \mathbb{R}^n$ if and only if $\nabla f_\ell(\boldsymbol{x})^\top \boldsymbol{d} < 0$, for all $\ell = 1, \ldots, m$.*

The next theorem gives a first order necessary condition for $\overline{\boldsymbol{x}}$ to be a local weak efficient solution.

Theorem 2.2.7. *[110, Thm. 5.1 (ii)(a)] Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions for $\ell = 1, \ldots, m$. If $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a local weak efficient solution of Problem (2.21), then*

$$\operatorname{range}(J_{\boldsymbol{F}}(\overline{\boldsymbol{x}})) \cap (-\operatorname{int}(\mathbb{R}^m_+)) = \emptyset, \tag{2.25}$$

where $J_{\boldsymbol{F}}(\cdot)$ denotes the Jacobian matrix of $\boldsymbol{F}$ at $\overline{\boldsymbol{x}}$, given by $J_{\boldsymbol{F}}(\overline{\boldsymbol{x}}) = \begin{bmatrix} \nabla f_1(\overline{\boldsymbol{x}})^\top \\ \vdots \\ \nabla f_m(\overline{\boldsymbol{x}})^\top \end{bmatrix}$.

According to Fliege and Svaiter [90], points that satisfies (2.25) are defined as critical (or stationary) points of Problem (2.21). Note that when $m = 1$, the multiobjective optimization Problem (2.21) simplifies to the single-objective optimization Problem (2.1). Consequently, Pareto solutions to (2.21) coincide with solutions of (2.1). Moreover, the Pareto criticality condition (2.25) implies the stationarity condition (2.2).

The relationship between weak efficiency and criticality, as outlined in [89, Thm. 3.1], is provided in the following result.

Theorem 2.2.8. *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and convex functions for $\ell = 1, \ldots, m$. If $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a critical point of Problem (2.21), then $\overline{\boldsymbol{x}}$ is a weakly efficient solution.*

As seen in Section 2.1, a minimizer cannot have a descent direction. In multiobjective optimization, there is also a fundamental connection between Pareto criticality and descent directions. Considering Problem (2.21), if $\boldsymbol{x}$ is not considered a Pareto critical point, it implies the existence of a descent direction for $\mathbf{F}$ at $\boldsymbol{x}$. Conversely, Pareto critical points lack any descent direction for $\mathbf{F}$. A necessary condition for $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ to be a local Pareto minimizer is stated in the following proposition [90].

Proposition 2.2.9. *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions for $\ell = 1, \ldots, m$. If $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ is a local Pareto minimizer for Problem (2.21), then for all $\boldsymbol{d} \in \mathbb{R}^n$, there exist $\ell_{\boldsymbol{d}} \in \{1, \ldots, m\}$ such that $\nabla f_{\ell_{\boldsymbol{d}}}(\overline{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0$.*

Based on the previous result, Fliege and Svaiter [90, Lem. 3] proposed a criticality measure for multiobjective optimization, formalized in the following proposition.

Proposition 2.2.10. *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable functions for $\ell = 1, \ldots, m$. Define*

$$\mu(\boldsymbol{x}) := -\min_{\|\boldsymbol{d}\| \leq 1} \max_{\ell=1,\ldots,m} \nabla f_\ell(\boldsymbol{x})^\top \boldsymbol{d}. \tag{2.26}$$

The following statements hold:

- *$\mu(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$;*
- *if $\boldsymbol{x} \in \mathbb{R}^n$ is a Pareto critical point of Problem (2.21) then $\mu(\boldsymbol{x}) = 0$ and $\mathbf{0} \in \mathcal{M}(\boldsymbol{x})$, where $\mathcal{M}(\boldsymbol{x})$ denotes the solution set of (2.26);*
- *if $\boldsymbol{x} \in \mathbb{R}^n$ is not a Pareto critical point of Problem (2.21) then $\mu(\boldsymbol{x}) > 0$ and for any $\boldsymbol{d} \in \mathcal{M}(\boldsymbol{x})$ we have*

  $$\nabla f_j(\boldsymbol{x})^\top \boldsymbol{d} \leq \max_{\ell=1,\ldots,m} \nabla f_\ell(\boldsymbol{x})^\top \boldsymbol{d} < 0, \ \forall j \in \{1, \ldots, m\},$$

  i.e. $\boldsymbol{d}$ is a descent direction of Problem (2.21);
- *the function $\boldsymbol{x} \mapsto \mu(\boldsymbol{x})$ is continuous;*
- *if $\boldsymbol{x}_k$ converges to $\overline{\boldsymbol{x}}$, $\boldsymbol{d}_k \in \mathcal{M}(\boldsymbol{x}_k)$, and $\boldsymbol{d}_k$ converges to $\overline{\boldsymbol{d}}$, then $\overline{\boldsymbol{d}} \in \mathcal{M}(\overline{\boldsymbol{x}})$.*

Function μ provides insights into the Pareto criticality of a specific point, functioning similarly to the norm of the gradient in single-objective optimization. Custódio et al. [63, Lem. 2.2] stated the relationship between the function μ and the gradient of each objective component.

Lemma 2.2.11. *For a given $\boldsymbol{x} \in \mathbb{R}^n$ and $\varepsilon > 0$, assume that $\nabla f_\ell(x)$ is well defined for $\ell = 1, \ldots, m$. If $\mu(\boldsymbol{x}) > \varepsilon$, then $\|\nabla f_\ell(\boldsymbol{x})\| > \varepsilon$, for all $\ell = 1, \ldots, m$.*

Note that, by the contrapositive of Lemma 2.2.11, for a given $\boldsymbol{x} \in \mathbb{R}^n$, if there exists $\ell \in \{1, \ldots, m\}$ such that $\|\nabla f_\ell(\boldsymbol{x})\| \leq \varepsilon$, then we can conclude that $\mu(\boldsymbol{x}) \leq \varepsilon$.

We now assume that the feasible region of Problem (2.21) is defined by a set of general constraints expressed as

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0} \text{ and } \boldsymbol{h}(\boldsymbol{x}) = \mathbf{0}\}$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$, and $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^q$. Additionally, we assume that the functions f_ℓ (for $\ell = 1, \ldots, m$), g_i (for $i = 1, \ldots, p$), and h_j (for $j = 1, \ldots, q$) are continuously differentiable. Let $I(\boldsymbol{x})$ denote the set of indices corresponding to active inequalities at a feasible point $\boldsymbol{x} \in \Omega$:

$$I(\boldsymbol{x}) = \{i \in \{1, \ldots, p\} \mid g_i(\boldsymbol{x}) = 0\}.$$

Again, as in the single-objective case, to derive the Fritz-John necessary optimality conditions for Problem (2.21), we present the following auxiliary result.

Lemma 2.2.12. *If $\overline{x} \in \Omega$ is an efficient solution for the Problem (2.21), then the system*

$$\begin{aligned} &\nabla f_\ell(\overline{x})^\top \boldsymbol{d} < 0, \ \ell = 1, \ldots, m \\ &\nabla g_i(\overline{x})^\top \boldsymbol{d} \leq 0, \ i \in I(\overline{x}) \\ &\nabla h_j(\overline{x})^\top \boldsymbol{d} = 0, \ j = 1, \ldots, q \end{aligned} \tag{2.27}$$

has no solution $\boldsymbol{d} \in \mathbb{R}^n$.

By applying Motzkin's Theorem to the system (2.27), we derive the Fritz-John necessary optimality conditions for Problem (2.21). This result extends Theorem 2.1.17 to the multiobjective optimization problem (2.21), and can be found in [67, Thm. 69].

Theorem 2.2.13 (Fritz-John necessary optimality conditions)**.** *If $\overline{x} \in \Omega$ is an efficient solution for Problem (2.21) then there exist vectors $\boldsymbol{\theta} \in \mathbb{R}^m_+$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, and $\boldsymbol{\mu} \in \mathbb{R}^q$, with $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \neq \mathbf{0}$ such that*

$$\sum_{\ell=1}^{m} \theta_\ell \nabla f_\ell(\overline{x}) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\overline{x}) = 0, \tag{2.28}$$

$$\lambda_i g_i(\overline{x}) = 0, \quad i \in \{1, \ldots, p\}. \tag{2.29}$$

Under the Guignard Constraint Qualification (GCQ), it is possible to ensure that the vector of Lagrange multipliers associated with the objective function components is non null. For more details, see [142, Thm. 2.1] or [177, Thm. 3.1].

Theorem 2.2.14 (KKT necessary optimality conditions)**.** *Let $\overline{x} \in \Omega$ be an efficient solution for Problem (2.21) and suppose that (GCQ) holds. Then, there exist vectors $\boldsymbol{\theta} \in \mathbb{R}^m_+ \backslash \{\mathbf{0}\}$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, and $\boldsymbol{\mu} \in \mathbb{R}^q$ such that*

$$\sum_{\ell=1}^{m} \theta_\ell \nabla f_\ell(\overline{x}) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\overline{x}) = 0, \tag{2.30}$$

$$\lambda_i g_i(\overline{x}) = 0, \quad i \in \{1, \ldots, p\}. \tag{2.31}$$

The same result was obtained by Da Cunha and Polak [67, Corr. 76] under the linear independence constraint qualification (LICQ), i.e., the set $\{\nabla g_i(\overline{x})\}_{i \in I(\overline{x})} \cup \{\nabla h_j(\overline{x})\}_{j=1,\ldots,q}$ is linearly independent.

As in the single-objective case, under the assumption of convexity, the KKT conditions are not only necessary but also sufficient. Thus, under the assumption of convexity, if a point $\overline{x} \in \Omega$ satisfies the system (2.30)-(2.31), then $\overline{x}$ is an efficient solution for Problem (2.21) (see [177, Thm. 3.2]).

Theorem 2.2.15 (KKT sufficient optimality conditions)**.** *Let $\overline{x} \in \Omega$, and f_ℓ (for $\ell = 1, \ldots, m$), g_i (for $i = 1, \ldots, p$), and h_j ($j = 1, \ldots, q$) be convex functions. If $\overline{x} \in \Omega$ satisfies (2.30)-(2.31) with $\boldsymbol{\theta} \in \mathbb{R}^m_+ \backslash \{\mathbf{0}\}$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, and $\boldsymbol{\mu} \in \mathbb{R}^q$ then $\overline{x}$ is an efficient solution for Problem (2.21).*

All the previously results assumed that the involved functions are continuously differentiable. As we have done in the last section, we will also address the case where the functions involved are nonsmooth, using Clarke's analysis.

In [18, Thm. 2.4], a necessary condition for Pareto optimality is provided, that can be regarded as the counterpart of Theorem 2.2.7 for the nonsmooth case.

Theorem 2.2.16 (Pareto-Clarke necessary optimality conditions)**.** *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be Lipschitz continuous for $\ell = 1, \ldots, m$ near $\overline{\boldsymbol{x}} \in \Omega$. If $\overline{\boldsymbol{x}}$ is a local Pareto point of Problem (2.21), then for any $\boldsymbol{d} \in T_\Omega^{Cl}(\overline{\boldsymbol{x}})$ there exists some $\ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\}$ such that*

$$f_\ell^\circ(\overline{\boldsymbol{x}}; \boldsymbol{d}) \geq 0.$$

Due to Proposition 2.2.9, the following corollary is derived, as stated in [18, Cor. 2.5].

Corollary 2.2.17 (Pareto-Clarke-KKT necessary optimality conditions)**.** *Let $f_\ell : \mathbb{R}^n \to \mathbb{R}$ be strictly differentiable for $\ell = 1, \ldots, m$ at $\overline{\boldsymbol{x}} \in \Omega$. If $\overline{\boldsymbol{x}}$ is a local Pareto point of Problem (2.21), then for any $\boldsymbol{d} \in T_\Omega^{Cl}(\overline{\boldsymbol{x}})$ there exists some $\ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\}$ such that*

$$\nabla f_\ell(\overline{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0.$$

As we have done in the last section, we will now consider the feasible region for Problem (2.21) to be defined by

$$\Omega = \{\boldsymbol{x} \in X \mid \boldsymbol{g}(\boldsymbol{x}) \leq 0\},$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^p$ and $X \subseteq \mathbb{R}^n$ is a nonempty set. Moreover, we assume that f_ℓ ($\ell = 1, \ldots, m$) and g_i ($i = 1, \ldots, p$) are locally Lipschitz continuous.

The following results were established by Liuzzi, Lucidi, and Rinaldi in [134], taking into account the set of feasible directions $\mathcal{F}_X(\boldsymbol{x})$ of the set X when this set defined by $X := \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{lb} \leq \boldsymbol{x} \leq \mathbf{ub}\}$, with $\mathbf{lb} < \mathbf{ub}$.

Proposition 2.2.18. *[134, Pareto-Clarke Fritz-John necessary optimality conditions, Prop. 3.3] Let $\overline{\boldsymbol{x}} \in \Omega$ be a local Pareto point for Problem (2.21). Then, there exist multipliers $\boldsymbol{\theta} \in \mathbb{R}_+^m$, $\boldsymbol{\lambda} \in \mathbb{R}_+^p$, not all zero, and such that $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ for $i = 1, \ldots, p$, and a vector*

$$\boldsymbol{v} \in \sum_{\ell=1}^{m} \theta_\ell \partial f_\ell(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{\boldsymbol{x}}),$$

such that

$$\boldsymbol{v}^\top \boldsymbol{d} \geq 0 \quad \text{for every } \boldsymbol{d} \in \mathcal{F}_X(\boldsymbol{x}).$$

Assuming that $\overline{\boldsymbol{x}} \in \Omega$ satisfies (EMFCQ), we can establish the KKT optimality conditions for Problem (2.21).

Proposition 2.2.19. *[134, Pareto-Clarke KKT necessary optimality conditions, Prop. 3.4] Let $\overline{\boldsymbol{x}} \in \Omega$ be a local Pareto point for Problem (2.21) and assume that $\overline{\boldsymbol{x}}$ satisfies (EMFCQ). Then, there exist multipliers $\boldsymbol{\theta} \in \mathbb{R}^m_+ \backslash \{\mathbf{0}\}$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, such that $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ for $i = 1, \ldots, p$, and a vector*

$$\boldsymbol{v} \in \sum_{\ell=1}^{m} \theta_\ell \partial f_\ell(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{\boldsymbol{x}}),$$

such that

$$\boldsymbol{v}^\top \boldsymbol{d} \geq 0 \quad \text{for every } \boldsymbol{d} \in \mathcal{F}_X(\boldsymbol{x}).$$

The previous optimality condition motivates the following definition of Pareto-Clarke stationary point for Problem (2.21) when $\Omega = \{\boldsymbol{x} \in X \mid \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}\}$.

Definition 2.2.20 (Pareto-Clarke stationary point)**.** *A point $\overline{\boldsymbol{x}} \in \Omega$ is a Pareto-Clarke stationary point for Problem (2.21) if there exist multipliers $\boldsymbol{\theta} \in \mathbb{R}^m_+ \backslash \{\mathbf{0}\}$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, satisfying $\lambda_i g_i(\overline{\boldsymbol{x}}) = 0$ for $i = 1, \ldots, p$, and a vector*

$$\boldsymbol{v} \in \sum_{\ell=1}^{m} \theta_\ell \partial f_\ell(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \lambda_i \partial g_i(\overline{\boldsymbol{x}}),$$

such that

$$\boldsymbol{v}^\top \boldsymbol{d} \geq 0 \quad \text{for every } \boldsymbol{d} \in \mathcal{F}_X(\boldsymbol{x}).$$

While the theory of optimality conditions for multiobjective problems with inequality constraints is well established in the nonsmooth case, the study of multiobjective problems involving both inequality and equality constraints has garnered increasing interest. It is worth noting that equality constraints can be reformulated as two inequality constraints, which provides additional flexibility in the analysis and allows the use of well-established results for problems only with inequality constraints. In recent years, considerable efforts have been devoted to formalizing these conditions, aiming to enhance the understanding and applicability of optimality conditions in this more general context (see for instance [151]).

This chapter elucidated on several optimality conditions for both single-objective and multiobjective optimization problems, highlighting the existing connections. The next chapter will delve into various methodologies aimed at numerically solving single and multiobjective optimization problems.

3

Optimization Algorithms

In this chapter, we will describe several optimization methods with or without resource to the derivative information, specifically focusing on their relationship with the type of constraints present and whether the problem is single-objective (Section 3.1) or multiobjective (Section 3.2).

3.1 Single-objective optimization

Consider an unconstrained problem, meaning $\Omega = \mathbb{R}^n$ in Problem (2.1), and assume that derivative information is available

A basic algorithm will iteratively seek a sufficient decrease of the current function value, using a descent direction (see Definition 2.1.7). Algorithm 1 formalizes the procedure.

Algorithm 1 A Basic Line-Search Descent Direction Algorithm

Initialization: Choose an initial point $\boldsymbol{x}_0 \in \mathbb{R}^n$.

For $k = 0, 1, 2, \ldots$

Step 1: Compute a descent direction $\boldsymbol{d}_k$ at $\boldsymbol{x}_k$, meaning $\nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d}_k < 0$.

Step 2: Choose a stepsize $\alpha_k \in \mathbb{R}_+$ that guarantees a sufficient decrease for f at $\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$.

Step 3: Update the iterate point $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$ and set $k := k + 1$.

EndFor

Depending on the type of descent direction considered, Algorithm 1 instantiates in different optimization methods. The Steepest Descent method uses $\boldsymbol{d}_k = -\nabla f(\boldsymbol{x}_k)$. Newton's method considers first and second-order information to compute $\boldsymbol{d}_k = -(\nabla^2 f(\boldsymbol{x}_k))^{-1} \nabla f(\boldsymbol{x}_k)$. In general, this latter approach is computationally more efficient than steepest descent, but it can also be computationally more expensive due to the need to calculate and store the Hessian matrix. Additionally, it may encounter difficulties when the Hessian is not positive

definite, as this could result in a direction that does not provide descent. The simplicity of first-order methods and efficiency of second-order approaches are combined in quasi-Newton methods. These are extensions of Newton's method that approximate the Hessian matrix without explicitly computing it. In this class of optimization methods, the direction is given by $\boldsymbol{d}_k = -H(\boldsymbol{x}_k)^{-1}\nabla f(\boldsymbol{x}_k)$, where $H(\boldsymbol{x}_k)$ approximates $\nabla^2 f(\boldsymbol{x}_k)$. Two examples are the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm and its limited-memory variant (L-BFGS). More details can be found in [152].

In the line-search strategy, more precisely, in **Step 2** of Algorithm 1, the ideal choice for α_k would be the global solution of $\min_{\alpha \in \mathbb{R}_+} f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$. However, finding this global solution is not always simple, and sometimes even impossible. Therefore, instead of computing the true global minimum, the line-search approximates it within a limited number of trial stepsizes, by imposing the satisfaction of a sufficient decrease condition for accepting a new point. Once this new point is found, a new search direction and stepsize are computed at the updated point, and the process is repeated [152].

To ensure the convergence of an algorithm based on line-search and descent directions, it is necessary to assume that the cosine measure between the descent direction and the gradient at the current point is sufficiently large. This angle condition ensures that the descent direction considered is not orthogonal to the gradient and thus makes noticeable progress toward a solution. In the next theorem, we state the result that establishes the global convergence for Algorithm 1.

Theorem 3.1.1. *[109, Thm. 11.7] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function and $\boldsymbol{x}_0 \in \mathbb{R}^n$ be an initial point. Assume that the level set $\mathcal{L}(\boldsymbol{x}_0) = \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ is bounded and ∇f is Lipschitz continuous on an open domain containing $\mathcal{L}(\boldsymbol{x}_0)$. Let $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ be the sequence of iterates generated by Algorithm 1, where $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$, with the vectors $\boldsymbol{d}_k$ satisfying $\cos(\theta) = -\frac{\boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}_k\|\|\nabla f(\boldsymbol{x}_k)\|} \geq \varepsilon > 0$ and the scalar α_k chosen as the first element of the sequence $\{1, \frac{1}{2}, \frac{1}{3}, \ldots\}$ that satisfies the sufficient decrease condition $f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + \mu\alpha_k \boldsymbol{d}_k^\top \nabla f(\boldsymbol{x}_k)$, where $\mu \in]0, 1[$ (known as the Armijo condition). Then*

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

For more details, see [152, Ch. 3].

Differently from the algorithms previously described, a Trust-Region Method repeatedly updates a model of the objective function, inside a certain region surrounding the current iterate point, known as the trust region. The model considered should be an accurate approximation of the function inside the trust region. Typical approaches consider linear or quadratic Taylor expansions of the objective function. By using derivative information, the algorithm finds a step within the trust region that optimizes the model at each iteration. Algorithm 2 presents a basic trust-region algorithm.

At each iteration, the model function m_k is calculated around the current iterate $\boldsymbol{x}_k$ and minimized within the trust region Δ_k (**Step 1**). Depending on how accurately the model reduction predicts the actual decrease in the objective function value (**Step 2**), the

acceptance or rejection of the model minimizer occurs (**Step 3**), and the trust-region radius is adjusted accordingly (**Step 4**). For further details, refer to [98, 109, 152].

Algorithm 2 A Basic Trust-Region Algorithm

Initialization: Choose an initial point $\boldsymbol{x}_0 \in \mathbb{R}^n$, $\overline{\Delta} > 0$, $\Delta_0 \in (0, \overline{\Delta}]$, and $0 < \mu < \eta < 1$.

For $k = 0, 1, 2, \ldots$

Step 1: Consider a model m_k of f centered at x_k and compute the direction $\boldsymbol{d}_k = \arg\min_{\|\boldsymbol{d}\| \leq \Delta_k} m_k(\boldsymbol{d})$.

Step 2: Compute the reduction indicator $\rho_k = \frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + \boldsymbol{d}_k)}{m_k(\boldsymbol{0}) - m_k(\boldsymbol{d}_k)} = \frac{\text{Actual reduction}}{\text{Predicted reduction}}$.

Step 3: If $\rho_k \leq \mu$, then $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ (unsuccessful iteration). Otherwise, update the iterate point $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \boldsymbol{d}_k$ (successful iteration).

Step 4: Update the trust-region radius $\Delta_{k+1} := \begin{cases} \frac{1}{2}\Delta_k & \text{if } \rho_k \leq \mu \\ \Delta_k & \text{if } \mu < \rho_k < \eta, \\ \min\{2\Delta_k, \overline{\Delta}\} & \text{if } \rho_k \geq \eta \end{cases}$ and set $k := k + 1$.

EndFor

The global convergence of Algorithm 2 is stated on the following theorem.

Theorem 3.1.2. *[109, Thm. 11.11] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function and $\boldsymbol{x}_0 \in \mathbb{R}^n$ be an initial point. Assume that the level set $\mathcal{L}(\boldsymbol{x}_0) = \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ is bounded and $f \in C^2$, i.e., f, ∇f, and $\nabla^2 f$ are continuous in an open set containing $\mathcal{L}(\boldsymbol{x}_0)$. Let $\{\boldsymbol{x}_k\}_{k \in \mathbb{N}}$ be the sequence of iterates generated by Algorithm 2. Then*

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

Consider now constrained optimization problems, where the feasible region is not the entire space. Constraints can be of several types, including linear and nonlinear constraints, integer constraints, binary constraints, mixed constraints, categorical constraints, and so on. In this thesis, the main focus will be on linear and nonlinear constraints. Specifically, concerning linear constraints, we will distinguish between bounds and general linear constraints.

Bound constraints shape the feasible region by defining permissible ranges for decision variables, corresponding to practical and meaningful real-world limitations on the variables. In this case, the feasible region defined for Problem (2.1) is given by

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{lb} \leq \boldsymbol{x} \leq \mathbf{ub}\},$$

where $\mathbf{lb} < \mathbf{ub}$ correspond to lower and upper bounds on the variables, respectively.

Although bound constraints are conceptually simple, their presence in an optimization problem can introduce significant complexities, as they can affect the effective search for optimal solutions, since iterates should remain feasible.

To this end, some algorithms adopt strategies such as the direct projection of points into the bounds, defined as follows:

$$P(\boldsymbol{x},\ \mathbf{lb},\ \mathbf{ub})_i = \begin{cases} lb_i & \text{if } x_i < lb_i, \\ x_i & \text{if } x_i \in [lb_i, ub_i] \\ ub_i & \text{if } x_i > ub_i \end{cases} \tag{3.1}$$

assuming, without loss of generality, that $\mathbf{lb} < \mathbf{ub}$. Note that $P(\boldsymbol{x},\ \mathbf{lb},\ \mathbf{ub})$ can also be expressed as $P(\boldsymbol{x},\ \mathbf{lb},\ \mathbf{ub}) = \max\{\ \mathbf{lb}, \min\{\boldsymbol{x},\ \mathbf{ub}\}\}$, where the maximum and minimum are taken component-wise.

In [53], an extension of a trust region algorithm for unconstrained optimization was proposed to accommodate bounds on the variables, assuming that the objective function is twice continuously differentiable and employing the projection into bounds, as defined in (3.1).

Consider now general linear constraints, i.e., that the feasible region is defined by

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{r \times n}$ and $\boldsymbol{b} \in \mathbb{R}^r$.
As for bound constraints, projections can also be performed for general linear constraints. However, the process is not as straightforward as outlined in the closed formula (3.1). Instead, a projection of a point $\boldsymbol{x} \in \mathbb{R}^n$ in Ω can be obtained by solving the following minimization problem

$$\begin{aligned} \overline{x} \in \arg\min\ & \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \\ \text{s.t. } & \boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b} \end{aligned} \tag{3.2}$$

Figure 3.1 provides an illustration depicting the projection of an initially infeasible point $\boldsymbol{x}$ into a feasible region Ω, defined only considering linear constraints.

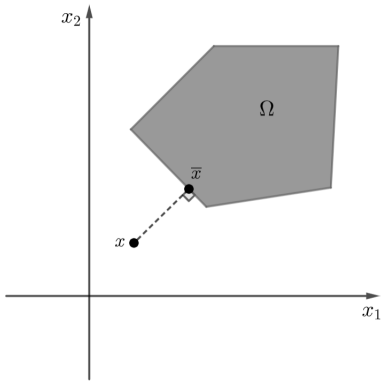


Figure 3.1: Example of the projection of point $\boldsymbol{x}$ into the feasible region Ω, defined only using linear constraints.

The projection technique was used to generalize methods well-known in unconstrained optimization to linearly constrained problems, such as the projected gradient method (see [41]).

In addition, other strategies have been developed to address this type of problem, such as active set algorithms. This method iteratively updates a subset of constraints, referred to as the active set, which are treated as binding at the current iterate. At each iteration, a subproblem is solved within the subspace defined by the active constraints to determine the optimal step direction. If the resulting point satisfies specific optimality conditions, such as feasibility and stationarity, it is accepted as the new iterate. Otherwise, the active set is updated by adding newly active constraints or removing those that are no longer binding. This iterative process continues until a termination criterion, is satisfied (see [98, Sec. 5.2]). Active set methods are particularly effective for problems where the number of active constraints remains relatively small compared to the total number of constraints, as this can significantly reduce computational costs.

Until now, only linear inequalities have been addressed. When in presence of linear equalities, by expressing certain variables in terms of others, the dimension of the optimization problem can be significantly reduced. This reduction not only simplifies the problem but also enhances computational efficiency. However, as mentioned in [152], the elimination of variables may not always be easy or yield advantages when inequality constraints coexist with equalities. Therefore, it is crucial to have caution in employing elimination techniques, as they have the potential to modify the problem or introduce ill-conditioning.

The last type of constraints that we will address are the nonlinear constraints. Consider a general feasible region, defined by

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{g}(\boldsymbol{x}) \leq \mathbf{0} \text{ and } \mathbf{h}(\boldsymbol{x}) = \mathbf{0}\},$$

where $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^p$ and $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^q$, are nonlinear functions.

If Ω can be regarded as an intersection of a finite number of convex sets where it is simple to project, we can apply Dykstra's algorithm to project a point into Ω (for more details see [35, 82]). In [33], Birgin, Martínez, and Raydan proposed a spectral projected gradient method for addressing constrained optimization problems, where the feasible region Ω is a closed convex set. Within each iteration of this method, a step is taken along the feasible projected gradient direction $\boldsymbol{d}_k = P_\Omega(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k)) - \boldsymbol{x}_k$, where $\alpha_k > 0$ is the widely recognized spectral choice of stepsize, and $P_\Omega(\boldsymbol{z})$ represents the Euclidean projection of $\boldsymbol{z}$ onto Ω. In [32], Dykstra's alternating projection algorithm [35] is used to obtain a direction $\boldsymbol{d}_k$ such that $\boldsymbol{x}_k + \boldsymbol{d}_k \in \Omega$.

A common approach to handle general nonlinear constraints is to introduce penalties in the objective function, representing violations of the constraints. In this kind of exterior approach, the algorithm moves from outside the feasible region towards a feasible solution.

For instance, we can consider the penalty function given by

$$Z(\boldsymbol{x};\rho) = f(\boldsymbol{x}) + \frac{1}{\rho}\left(\sum_{i=1}^{p} \max\{0, g_i(\boldsymbol{x})\}^{\nu} + \sum_{j=1}^{q} \left|h_j(\boldsymbol{x})\right|^{\nu}\right) \tag{3.3}$$

where $\nu > 1$ is a smoothing term and $\rho > 0$ a penalty parameter. However, the appropriate choice of penalty parameter ρ is crucial, as very high values can lead to overly conservative approaches, while low values may not be effective in generating solutions satisfying the desired constraints (for more details see [62, ch. 9]).

Without the smoothing term ν in the penalty function, this function does not have first-order derivatives defined everywhere. Therefore, the gradient-type methods previously described are not usually applied to the unconstrained subproblems needed to be solved at each iteration. Taking into account the original penalty function (3.3) with $\nu = 1$, Conn proposed an algorithm in [52] that employs a Quasi-Newton method applied to a perturbed version of this penalty function. Unlike the previous algorithm using only one penalty parameter, Lasserre [127] proposed an algorithm to solve a constrained problem using an exact penalty function similar to (3.3) but with one penalty parameter for each constraint function. This algorithm adjusts the penalty parameters based on a test function to prevent constraints with small associated Lagrange multipliers from being overly penalized.

An alternative approach for handling general constraints is to evaluate only feasible points and assign the value $+\infty$ when a point is infeasible. Consequently, we can define the following extreme barrier function:

$$f_{\Omega}(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \Omega \\ +\infty & \text{otherwise} \end{cases}. \tag{3.4}$$

During the optimization process, f_{Ω} is considered instead of f.

Note that this approach does not distinguish between violations of inequalities or equalities constraints. To overcome this situation a logarithmic barrier approach can be employed, resulting in the following merit function:

$$Z(\boldsymbol{x};\rho) = f(\boldsymbol{x}) - \rho \sum_{i=1}^{p} \log(-g_i(\boldsymbol{x})) + \frac{1}{\rho} \sum_{j=1}^{q} \left|h_j(\boldsymbol{x})\right|^{\nu}, \tag{3.5}$$

where $\nu > 1$ is a smoothing term, and $\rho > 0$ a penalty parameter. Barrier methods maintain feasibility by establishing a barrier that prevents the iterates from approaching the boundary of the feasible region.

In [26], the authors propose a method inspired by [85], which utilizes Newton-type directions. They employ the same mixed penalty approach described in equation (3.5), but two distinct penalty parameters instead of the single parameter ρ. These two penalty coefficients induce a two-parameter differentiable trajectory, enabling a monotonically decreasing sequence of errors on the unconstrained minimizers. Specifically, the penalty

parameter is decreased when the Euclidean norm of the gradient of the penalty function falls below the penalty parameter itself. For more details about interior methods see [92].

In 1969, the Augmented Lagrangian method was independently proposed by Hestenes [114] and Powell [158] to address constrained optimization problems. The method introduces a penalty term in the objective function, thereby also transforming a constrained optimization problem into an unconstrained one and allowing the constraints to be directly incorporated into the optimization process. The Augmented Lagrangian function associated with Problem (2.1) is defined by

$$L_\rho(\boldsymbol{x}, \lambda, \mu) = f(\boldsymbol{x}) + \frac{\rho}{2}\left[\sum_{i=1}^{p} \max\left\{0, g_i(\boldsymbol{x}) + \frac{\lambda_i}{\rho}\right\}^2 + \sum_{j=1}^{q}\left(h_j(\boldsymbol{x}) + \frac{\mu_j}{\rho}\right)^2\right]$$

for all $\boldsymbol{x} \in \mathbb{R}^n$, $\rho > 0$, $\boldsymbol{\lambda} \in \mathbb{R}^p_+$, and $\boldsymbol{\mu} \in \mathbb{R}^q$. By iteratively updating the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and the penalty parameter ρ, the Augmented Lagrangian method effectively balances between satisfying the constraints and optimizing the objective function. Its performance in handling large-scale problems is remarkable, and it finds applications across a broad range of fields.

A well-known algorithm based on the Augmented Lagrangian function that can address exceedingly large problems within reasonable computational time is `ALGENCAN` [4]. It employs the `GENCAN` algorithm, as proposed in [30], to address the minimization subproblems at each iteration. `GENCAN` was specifically designed for minimizing smooth functions, characterized by a potentially large number of variables and box constraints. Further details about the Augmented Lagrangian method can be found in [31].

An alternative strategy that bypasses the use of penalty or merit functions is the filter approach, initially proposed by Fletcher and Leyffer [87]. In this methodology, a measure of feasibility is considered as an additional objective that needs to be minimized. The original single-objective minimization problem is replaced by the biobjective minimization of $(f(\boldsymbol{x}),h(\boldsymbol{x}))$, where $h(\boldsymbol{x})$ measures how close the point $\boldsymbol{x}$ is from being feasible. Subsequently, it avoids the acceptance of dominated points, resorting to Pareto dominance. This enables the acceptance of a new point if it leads to a reduction in either the objective function f, or the constraint violation function, h.

Several filter-based algorithms can be found in the literature. In [103], a filter algorithm with global convergence to stationary points was proposed, where each iteration is composed of two independent phases and the filter provides the coupling between them. One of the phases is related to feasibility, reducing a measure of infeasibility given by the constraint violation function h. The other phase is related to optimality, reducing the objective function in a tangential approximation of the feasible region. In [86], the authors propose a trust-region SQP-filter algorithm for general nonlinear programming. This algorithm decomposes the step into normal and tangential components, enabling an approximate solution of the quadratic subproblem solved by the Newton's method.

Under standard assumptions, the sequence of iterates has at least one first-order critical limit point, ensuring global convergence. A survey of filter methods can be found in [88].

It is worth highlighting that the exterior approach, the augmented Lagrangian methods, and the filter strategy are particularly useful when the initialization provided is an infeasible point. The extreme barrier approach requires a feasible initialization.

From all that has been stated, it is clear that derivatives are an important tool in nonlinear optimization, since they can guide the search, by identifying directions of potential descent for the functions. However, several reasons could explain their absence. For instance, the problem by itself could be nonsmooth or, being smooth, could be contaminated by numerical noise, which would prevent the use of numerical techniques to estimate derivatives. Derivative-free optimization problems are often associated to black-box functions, in the context of simulation based optimization [14, 60]. In this situation, function evaluation is performed by running a typically expensive numerical simulation, which, provided some input corresponding to the problem variables, only returns the corresponding function value.

In single-objective derivative-free optimization, there are three main classes of methods [60]: direct search, unidirectional search, and trust-region methods. We will detail the direct search class, since it is the one closely related with the algorithms considered in the current thesis.

Direct search methods conduct the search for the best value of the objective function by sampling it in a finite set of points with good geometric characteristics. These methods do not attempt, either explicitly or implicitly, to approximate derivatives or to construct surrogate models for the objective function.

The algorithms belonging to this class can be divided into two subclasses: directional direct search, such as Generalized Pattern Search (GPS) [182] and Mesh Adaptive Direct Search (MADS) [13], and symplectic direct search, that limits the sampling to the vertices of a simplex, like is the case of Nelder-Mead [150].

Among direct search methods, directional direct search stands out. This algorithm class uses positive spanning sets as search directions. This special sets of vectors generate the hole space through nonnegative linear combinations of their elements. An important geometrical property is associated to positive spanning sets: every positive spanning set contains at least one element in every half-space of $\mathbb{R}^n$. Furthermore, if f is continuously differentiable ($f \in C^1$) and $\nabla f(\boldsymbol{x}) \neq 0$, then there is at least one element of any positive spanning set that is a descent direction.

In the context of pattern search methods, early works initially considered bound-constrained problems [129], utilizing the projection into the bound constraints as defined in (3.1). For general linear constraints, procedures were developed, inspired by the work of May [143], allowing the directions used by the algorithm to conform to the geometry of the feasible region imposed by the nearby constraints [122, 130, 138], and including specific strategies to address degenerate cases [2]. Moreover, in the presence of nonlinear equality constraints, augmented Lagrangian strategies have been adapted [128], reducing

the problem solution to a sequence of bound-constrained minimizations of an augmented Lagrangian function. Following the filter approaches proposed for derivative-based optimization [87], pattern search methods [10] have also incorporated this strategy to address inequality constrained problems.

In the original presentation of MADS [13], constraints were addressed with an extreme barrier approach (using the function defined in (3.4)), only evaluating feasible points. While this saves on function evaluations (a very relevant feature for the target problem class), it does not take advantage of the local information obtained about the feasible region. Based on the filter approach extended to pattern search in [10], linear and nonlinear inequalities started to be treated in MADS with the progressive barrier technique [11]. Later, the approach was extended to linear equality constraints [15], by reformulating the optimization problem, possibly reducing the number of original variables. All the improvements in MADS algorithm were implemented in the solver NOMAD [16].

For directional direct search methods using linesearch, other approaches have been taken to address general nonlinear constraints. These include nonsmooth exact penalty functions [133], where the original nonlinearly constrained problem is converted into the unconstrained or linearly constrained minimization of a nonsmooth exact penalty function. To overcome the limitations associated to this approach, in [136] a sequential penalty approach has been studied, based on the smoothing of the nondifferentiable exact penalty function, including a well-defined strategy for updating the penalty parameter. Recently, in the same algorithmic framework, Brilli *et. al* [39] proposed the use of a merit function (as defined in (3.5)) that handles inequality constraints by means of a logarithmic barrier approach and equality constraints by considering a penalization term. This strategy allows an easy management of relaxable and unrelaxable constraints and avoids the abrupt discontinuities introduced by the extreme barrier approach.

A basic directional direct search method is described in Algorithm 3.

Function $\overline{\rho}$ could be null or represent a forcing function $\rho : (0, +\infty) \to (0, +\infty)$, continuous and non-decreasing, that satisfies $\rho(t)/t \to 0$ when $t \downarrow 0$ (see [121]). Typical examples of forcing functions are $\rho(t) = \eta_1 t^{1+\eta_2}$, for $\eta_1, \eta_2 > 0$.

To establish the convergence of direct search methods of directional type, it is necessary to analyze the behavior of the algorithms at limit points of particular sequences of unsuccessful iterations. Therefore, the first step is to guarantee that $\liminf_{k \to +\infty} \alpha_k = 0$. Two different strategies can be adopted for it: imposing a sufficient decrease condition to accept new points or generating points in an implicit mesh, corresponding to an integer lattice. In the latter, $\overline{\rho}(\cdot) \equiv 0$ in Algorithm 3.

The following theorem states the convergence to zero of a subsequence of stepsizes, when using a globalization strategy based on sufficient decrease.

Theorem 3.1.3. *[121, Thm. 3.4] Let $\boldsymbol{x}_0 \in \Omega$ be an initial point and consider $f : \mathbb{R}^n \to \mathbb{R}$ a lower bounded function in the set $\mathcal{L}(\boldsymbol{x}_0) := \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$. Consider a globalization strategy based on imposing a sufficient decrease condition. Then, Algorithm 3 generates a sequence*

Algorithm 3 A Basic Directional Direct Search Algorithm - (DDS)

Initialization

Choose $x_0 \in \Omega$ such that $f(x_0) < +\infty$, an initial stepsize parameter $\alpha_0 \in \mathbb{R}_+$, $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, with directions d satisfying $0 < d_{\min} \leq \|d\| \leq d_{\max}$.

For $k = 0, 1, 2, \ldots$

1. **Search Step:** Compute a finite set of points $\{z_s\}_{s \in S}$ (in a mesh if $\bar{\rho}(\cdot) \equiv 0$). Evaluate f_Ω (as defined in (3.4)) at each element of $\{z_s\}_{s \in S}$ with the aim of identifying a point z that satisfies $f_\Omega(z) < f_\Omega(x_k) - \bar{\rho}(\alpha_k)$. If such a point is found, set $x_{k+1} = z$, declare the iteration successful, and skip the poll step.

2. **Poll step:** Select a set $D_k \in \mathcal{D}$. Evaluate f at the poll points $P_k = \{x_k + \alpha_k d \mid d \in D_k\}$, following a predetermined order. If a polling point $x_k + \alpha_k d$ is found such that $f\Omega(x_k + \alpha_k d) < f_\Omega(x_k) - \bar{\rho}(\alpha_k)$, then set $x_{k+1} = x_k + \alpha_k d$ and declare the iteration successful. Otherwise, declare the iteration unsuccessful and set $x_{k+1} = x_k$.

3. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k+1} = \gamma \alpha_k$. Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k+1} \in [\beta_1 \alpha_k, \beta_2 \alpha_k]$.

EndFor

of iterates satisfying

$$\liminf_{k \to +\infty} \alpha_k = 0.$$

Now, to state a similar result only using simple decrease to accept new points, the following assumption is required.

Assumption 3.1.4. *The set D of positive spanning sets is finite and the elements of D are of the form $G\bar{z}_j$, $j = 1, \ldots, |D|$, where $G \in \mathbb{R}^{n \times n}$ is a nonsingular matrix and each $\bar{z}_j$ is a vector in $\mathbb{Z}^n$.*

In the context of general constraints and potentially nonsmooth functions, it is necessary to consider an infinite set of directions $\mathcal{D}$, which must be dense (after normalization) in the unit sphere [13]. The set D serves as the basis for constructing the directions in $\mathcal{D}$.

Assumption 3.1.5. *Let D represent a finite set of positive spanning sets satisfying Assumption 3.1.4, with elements $d_k \in D_k \in \mathcal{D}$ obtained as nonnegative integer combinations of the columns of D.*

To comply with the integrality requirements, it is necessary to impose additional conditions when updating the stepsize parameter.

Assumption 3.1.6. *Let $\tau > 1$ be a rational number and $m^{\max} \geq 0$ and $m^{\min} \leq -1$ integers. If the iteration is successful, then the stepsize parameter is maintained or increased by considering*

$\alpha_{k+1} = \tau^{m^+}\alpha_k$ *with* $m^+ \in \{0, \ldots, m^{\max}\}$. *If the iteration is unsuccessful, then the stepsize parameter is decreased by setting* $\alpha_{k+1} = \tau^{m^-}\alpha_k$, *with* $m^- \in \{m^{\min}, \ldots, -1\}$.

In addition, the points generated at the search step need to lie in the implicit mesh considered at each iteration by the algorithm.

Assumption 3.1.7. *At iteration k, the search step in Algorithm 3 only evaluates points in*

$$M_k = \bigcup_{\boldsymbol{x} \in E_k} \{\boldsymbol{x} + \alpha_k D\boldsymbol{z} \mid \boldsymbol{z} \in \mathbb{N}_0^{|D|}\},$$

where E_k *represents the set of all points evaluated by the algorithm previously to iteration k.*

The following theorem establishes a result similar to the one of Theorem 3.1.3, but using a globalization strategy based on an implicit mesh.

Theorem 3.1.8. *[121, Thm 3.8] Let* $\boldsymbol{x}_0 \in \Omega$ *be an initial point and consider a function* $f : \mathbb{R}^n \to \mathbb{R}$. *Assume that the set* $\mathcal{L}(\boldsymbol{x}_0) := \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ *is compact. Under one of the Assumptions 3.1.4 or 3.1.5 combined with Assumptions 3.1.6 and 3.1.7, Algorithm 3 generates a sequence of iterates satisfying*

$$\liminf_{k \to +\infty} \alpha_k = 0.$$

The convergence analysis of directional direct search methods focuses on studying the algorithm's behavior at limit points of particular sequences of unsuccessful iterations, named as refining subsequences.

Definition 3.1.9. *A subsequence* $\{\boldsymbol{x}_k\}_{k \in K}$ *of iterates generated by Algorithm 3 corresponding to unsuccessful poll steps is said to be a refining subsequence if* $\{\alpha_k\}_{k \in K}$ *converges to zero. A limit point* $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ *of a refining subsequence is said to be a refined point.*

The concept of refining direction is associated with refined points and is formalized in Definition 3.1.10.

Definition 3.1.10. *Let* $\bar{\boldsymbol{x}}$ *be the refined point of a convergent refining subsequence* $\{\boldsymbol{x}_k\}_{k \in K}$. *If* $\lim_{k \in K'} \frac{\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|}$ *exists, where* $K' \subseteq K$, $\boldsymbol{d}_k \in \boldsymbol{D}_k$, *and* $\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \in \Omega$, *for sufficiently large* $k \in K'$, *then this limit is said to be a refining direction for* $\bar{\boldsymbol{x}}$.

In the next theorem, in a differentiable context, we state an upper bound for the norm of the gradient, directly related with the stepsize parameter.

Theorem 3.1.11. *[121, Thm. 3.3] Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be continuously differentiable, and suppose that* ∇f *is Lipschitz continuous with constant L. Then Algorithm 3 produces iterates such that for any unsuccessful iteration* $k \in \mathcal{U}$, *we have*

$$\|\nabla f(\boldsymbol{x}_k)\| \leq \kappa(\boldsymbol{D}_k)^{-1} \left[L\alpha_k d_{\max} + \frac{\bar{\rho}(\alpha_k)}{\alpha_k d_{\min}} \right],$$

where $\kappa(\boldsymbol{D}_k) = \min_{\boldsymbol{v} \in \mathbb{R}^n} \max_{\boldsymbol{d} \in \boldsymbol{D}_k} \frac{\boldsymbol{v}^\top \boldsymbol{d}}{\|\boldsymbol{v}\| \|\boldsymbol{d}\|}$.

Now, if the cosine measure $\kappa(\boldsymbol{D}_k)$ is appropriately controlled, the globalization strategies ensure a stationary result for the method. Since $\boldsymbol{D}_k$ consists of positive spanning sets, it follows that $\kappa(\boldsymbol{D}_k) > 0$. Moreover, assuming that the cosine measure is bounded below, thus limiting how poorly the search directions can be chosen, we can state the following convergence result.

Theorem 3.1.12. *[60, Thm. 7.3] Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and suppose that ∇f is Lipschitz continuous with constant L. Under the hypotheses required for at least one of the globalization strategies (Theorems 3.1.3 or 3.1.8), and assuming there exists $\kappa_{\min} > 0$ such that $\kappa(\boldsymbol{D}_k) \geq \kappa_{\min}$, for all $k \in \mathbb{N}$, Algorithm 3 produces iterates such that*

$$\liminf_{k \to +\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0.$$

Directional direct search (DDS) methods are well-suited for a broad range of optimization problems, including those involving nonsmooth functions. By exploring multiple directions, DDS methods mitigate the impact of nonsmoothness and can often bypass regions where the function lacks smooth structure. The following result establishes a key property of refined points, ensuring that any associated refined direction satisfies a non-negativity condition for the Clarke generalized directional derivative.

Theorem 3.1.13. *[12, Thm. 3.7] Let $\{\boldsymbol{x}_k\}_{k \in K}$ be a refining subsequence and $\overline{\boldsymbol{x}} \in \Omega$ be the corresponding refined point, generated by Algorithm 3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function near $\overline{\boldsymbol{x}}$. Under the hypotheses necessary for at least one of the globalization strategies (Theorems 3.1.3 or 3.1.8), for any refined direction associated with $\overline{\boldsymbol{x}}$, $f^\circ(\overline{\boldsymbol{x}}; \boldsymbol{d}) \geq 0$.*

Moreover, under the assumption that the set of refining directions is asymptotically dense in the Clarke tangent cone to Ω at limit points of refining subsequences, convergence to Clarke critical points or Clarke-KKT critical points can be established.

Theorem 3.1.14. *[12, Thm 3.13 and Cor. 3.14] Consider a refining subsequence $\{\boldsymbol{x}_k\}_{k \in K}$ converging to $\bar{\boldsymbol{x}} \in \Omega$. Assume that f is Lipschitz continuous near $\bar{\boldsymbol{x}}$ and that $H_\Omega^{Cl}(\bar{\boldsymbol{x}}) \neq \emptyset$. If the set of refining directions for $\bar{\boldsymbol{x}}$ is dense in $T_\Omega^{Cl}(\bar{\boldsymbol{x}})$, then $\bar{\boldsymbol{x}}$ is a Clarke critical point, i.e., for all $\boldsymbol{d} \in T_\Omega^{Cl}(\bar{\boldsymbol{x}})$, we have*

$$f^\circ(\bar{\boldsymbol{x}}; \boldsymbol{d}) \geq 0.$$

Furthermore, if f is strictly differentiable at $\bar{\boldsymbol{x}}$, then $\bar{\boldsymbol{x}}$ is a Clarke-KKT critical point, meaning that for all $\boldsymbol{d} \in T_\Omega^{Cl}(\bar{\boldsymbol{x}})$, we have

$$\nabla f(\bar{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0.$$

For a comprehensive understanding of directional direct search methods, more details are provided in [14, 60]. Additionally, in [79] a recent survey was conducted from a theoretical viewpoint, going beyond classical textbooks and addressing the presence of nonsmoothness, noise, and constraints. This survey highlights the key theoretical features of these methods.

Moving now to the class of unidirectional search methods, this approach involves searching in a single direction at each iteration, resembling derivative-based unidirectional search. However, the absence of derivative information introduces several challenges. For instance, it becomes impossible to compute descent directions or to assess the quality of a direction as being descent.

Perhaps the most well-known algorithm that belongs to the class of unidirectional search methods for derivative-free optimization is the implicit filter algorithm. Initially introduced by Bortz and Kelley [34], this algorithm performs a linesearch using a simplex gradient as search direction. The term *implicit filter* refers to its capability of optimizing noisy functions, a feature attributed to the use of simplex gradients.

Finally, the class of trust-region methods, that as been previously discussed in the derivative-based setting, approximates the objective function by a model, within a given region of interest [54, 60]. In a derivative-free context, the model cannot be based in Taylor expansions. Instead, polynomial interpolation, regression, or minimum Frobenius norm models are considered. The models are progressively enriched and updated, whenever the objective function is evaluated at new points. The remaining algorithmic structure follows Algorithm 2.

Several derivative-free algorithms have been proposed based on trust-region methods, where most of the time what differs is the technique used for model construction. To compute a well-defined model, it is essential to have a poised interpolating sample set. The concept of poisedness [60], which relates to the geometry of the sample set, describes how well the points in the sample set are distributed. Consequently, these methods typically include a geometry phase aimed at ensuring the adequacy of the sample set.

In [96], the authors introduce a trust-region algorithm that completely omits the geometry phase, arguing that disregarding the control of sample set geometry does not compromise the algorithm's efficiency or robustness. The numerical experiments conducted by the authors suggested that the approach remains competitive with other model-based algorithms that incorporate a geometry phase. In contrast, [175] emphasized the crucial role of maintaining a well-structured sample set geometry. The authors argued that when the model gradient is small, ensuring the model's sufficient poisedness is fundamental to the effectiveness of model-based derivative-free optimization algorithms.

In a derivative-free context, the works of Powell [157, 159, 160, 161, 162, 163, 164] and those of Conn, along with Gould, Toint, Scheinberg and Vicente [54, 55, 56, 57, 58, 59], stand out as pioneering contributions in formulating and developing these methods. Recently, Zhang [188] introduced `PRIMA`, a comprehensive package designed to solve general nonlinear optimization problems without requiring derivatives. This package serves as the reference implementation for Powell's DFO methods, including COBYLA [157] (for nonlinearly constrained problems), UOBYQA [164] (for unconstrained problems), NEWUOA [163] (for unconstrained problems), BOBYQA [162] (for bound-constrained problems), and LINCOA (for linearly constrained problems). Notably, Powell did not publish a dedicated paper to formally introduce the LINCOA algorithm.

3.1.1 Performance assessment in single-objective optimization

In order to assess the numerical performance of an algorithm it is common the use of performance [75] and data [147] 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 instance, for an accuracy τ the following convergence test could be adopted:

$$f_M - f(\boldsymbol{x}) \geq (1-\tau)(f_M - f_L), \tag{3.6}$$

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 (3.6) 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 $c(\boldsymbol{x}) > 10^{-4}$, where

$$c(\boldsymbol{x}) = \sum_{i=1}^{p} \max\{g_i(\boldsymbol{x}), 0\} + \sum_{j=1}^{q} |h_j(\boldsymbol{x})|.$$

The performance of solver $s \in \mathcal{S}$ can be measured by the fraction of problems in which the performance ratio $\dfrac{t_{p,s}}{\min\left\{t_{p,s'} : s' \in S\right\}}$ is at most φ, given by:

$$\rho_s(\varphi) = \frac{1}{|P|}\left|\left\{p \in P \mid \frac{t_{p,s}}{\min\left\{t_{p,s'} : s' \in S\right\}} \leq \varphi\right\}\right|.$$

In other words, the 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 solver s performs the best. Robustness, as the percentage of problems that the algorithm is able to solve, can be perceived for high values of φ.

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.

3.2 Multiobjective objective optimization

The previous section focused on the optimization of a single-objective function. In this section, we will consider a multiobjective optimization problem formalized by (2.21).

Multiobjective optimization is a challenging scientific domain, not only from an academic point of view but also due to its wide range of applications [3, 40, 111, 118, 155, 172].

Standard approaches to address multiobjective optimization problems include scalarization techniques [80, 144], which transform the multiobjective optimization problem into a single-objective one. This allows the solutions to the multiobjective problem to be obtained as solutions to a classical nonlinear optimization problem. However, scalarization requires careful selection of weights and can fail to capture all Pareto optimal solutions, potentially missing important trade-offs between objectives [68]. Additionally, each scalarization run typically yields a single solution, requiring multiple runs to approximate the Pareto front. Another common approach is the use of evolutionary algorithms [70] or other type of meta-heuristics [22]. In the former, a population of candidate solutions is iteratively improved through crossover, mutation, and selection operators. Despite their flexibility, these approaches lack general theoretical guarantees and do not scale well.

More recently, considerable research effort has been dedicated to extending iterative techniques, initially designed for single-objective optimization, to multiobjective optimization, offering a viable alternative to scalarization approaches and meta-heuristics. Fliege and Svaiter [90] proposed a steepest descent algorithm for unconstrained multiobjective optimization and a feasible descent direction method for constrained multiobjective optimization, with convergence guarantees and where neither ordering information nor weighting factors for the different objective functions components are assumed to be known. Under some classical differentiability assumptions, and considering some constraint qualifications, these methods are shown to converge to a point $\overline{x} \in \Omega$ satisfying $\mu(\overline{x}) = 0$, as defined in Proposition 2.2.10 (see also [78]).

Inspired by this work, extensions of classical descent algorithms for single-objective optimization started being proposed, including Newton [45, 77, 89, 101, 184], quasi-Newton [7, 140, 148, 156, 165, 166, 167, 169], conjugate gradient [100, 102, 153, 154], conditional gradient [8, 44], and projected gradient [25, 76, 84, 94, 95].These algorithms were designed to produce a single Pareto optimal solution for the problem. Usually, to generate more points, several different initializations are considered, in an attempt to approximate the Pareto front. However, initializing with different points does not guarantee that different Pareto optimal solutions will be generated, and generally does not produce spread and uniform fronts.

To overcome this limitation, inspired by [64], in recent years, descent methods have been proposed that explicitly manage a list of candidate solutions [50, 124, 125]. These methods start with a set of mutually nondominated points and for each point of this set carry out the computation of a descent direction and a line-search along it.

The general Front Descent (FD) algorithm, proposed in [125], updates the list of nondominated points L_k at each iteration in a two-phase procedure. For every point in the list that has not yet been identified as Pareto critical, meaning that $\theta_I(\boldsymbol{x}_c) < -\sigma$, where

$$\theta_I(\boldsymbol{x}) = \min_{\boldsymbol{d} \in \mathbb{R}^n} \max_{\ell \in I} \nabla f_\ell(\boldsymbol{x})^\top \boldsymbol{d} + \frac{1}{2}\|\boldsymbol{d}\|^2, \tag{3.7}$$

with $\sigma > 0$ and $I = \{1, \ldots, m\}$, a descent direction is computed for all components of the objective function. The direction $\boldsymbol{d}_c$, should be a steepest-descent-related direction [123], meaning that it satisfies the conditions $\max_{\ell \in I} \nabla f_\ell(\boldsymbol{x}_c)^\top v_D^I(\boldsymbol{x}_c) \leq -\Gamma_1 \|v^I(\boldsymbol{x}_c)\|^2$ and $\|v_D^I(\boldsymbol{x}_c)\| \leq \Gamma_2 \|v^I(\boldsymbol{x}_c)\|^2$ for $\Gamma_1, \Gamma_2 > 0$, with $v_D^I(\boldsymbol{x}_c)$ given by

$$\boldsymbol{d}_c = v_D^I(\boldsymbol{x}_c) = \arg\min_{\boldsymbol{d} \in \mathbb{R}^n} \max_{\ell \in I} \nabla f_\ell(\boldsymbol{x}_c)^\top \boldsymbol{d} + \frac{1}{2} \boldsymbol{d}^\top \boldsymbol{B}_\ell(\boldsymbol{x}_c) \boldsymbol{d}, \tag{3.8}$$

where $\boldsymbol{B}_\ell(\boldsymbol{x}_c)$ is an approximation to the Hessian of the objective function f_ℓ at point $\boldsymbol{x}_c$, and $v_D^I(\boldsymbol{x}_c) = v^I(\boldsymbol{x}_c)$, when $\boldsymbol{B}_\ell(\boldsymbol{x}_c)$ is the identity matrix. Otherwise, d_c is defined as the steepest descent direction, i.e.,

$$\boldsymbol{d}_c = v^I(\boldsymbol{x}_c) = \arg\min_{\boldsymbol{d} \in \mathbb{R}^n} \max_{\ell \in I} \nabla f_\ell(\boldsymbol{x}_c)^\top \boldsymbol{d} + \frac{1}{2} \|\boldsymbol{d}\|^2. \tag{3.9}$$

Moreover, a line-search is performed along $\boldsymbol{d}_c$ to compute the stepsize α_c^k. In the second phase, the objective space is explored, potentially enriching the list of nondominated points by performing line-searches along partial descent directions $I' \subset I$, focusing on subsets of components of the objective function with $\theta_{I'}(\boldsymbol{z}_c^k) < 0$ and $\boldsymbol{z}_c^k = \boldsymbol{x}_c^k + \alpha_c^k \boldsymbol{d}_c$. Whenever a new point is added, a filtering procedure removes all points that become dominated, ensuring that the list of solutions only keeps nondominated points. After exploring all the points in the list, the process is repeated until a stopping criterion is met. Algorithm 4, formalizes the procedure.

In [125], a convergence result was provided stating that the sequence of sets generated by Algorithm 4 asymptotically approaches stationarity for all their points. Before formalizing this result, we need the following definition regarding a new type of Pareto-stationarity measure for sets of mutually nondominated points.

Definition 3.2.1. *[125, Def. 5.12] Let $\mathbb{L}$ be the set of all sets $L \subseteq \mathbb{R}^n$ of mutually nondominated points w.r.t. function $\boldsymbol{F}$. Define the map $\Theta : \mathbb{L} \to \mathbb{R}$ as*

$$\Theta(L) = \min_{\boldsymbol{x} \in L} \theta(\boldsymbol{x}),$$

where $\theta(\boldsymbol{x}) = \theta_I(\boldsymbol{x})$, with $I = \{1, \ldots, m\}$ as in (3.7).

The following assumption is related to the order in which the points in Algorithm 4 are chosen from the list at the beginning of an iteration.

Assumption 3.2.2. *In Algorithm 4, at each iteration k, the first point to be processed in the for loop (before initiating Phase 1), belongs to the set $\arg\min_{\boldsymbol{x} \in L_k} \theta(\boldsymbol{x})$.*

Under Assumption 3.2.2, it is possible to show that the sequence of sets of nondominated points produced by Algorithm 4 converges to a set of Pareto critical points.

Proposition 3.2.3. *[125, Propo. 5.14] Let L_0 be a set of mutually nondominated points and $\boldsymbol{x}_0 \in L_0$ be a point such that the set $\mathcal{L}(\boldsymbol{x}_0) = \bigcup_{\ell=1}^m \{\boldsymbol{x} \in \mathbb{R}^n \mid f_\ell(\boldsymbol{x}) \leq f_\ell(\boldsymbol{x}_0)\}$ is compact. Let $\{L_k\}_{k \in \mathbb{N}}$ be*

Algorithm 4 Line-Search Front Descent Algorithm

Initialization: Choose an initial $L_0 \subset \Omega$ set of mutually nondominated points w.r.t. $\boldsymbol{F}$, $\alpha_0 > 0$, $\delta \in (0,1)$, $\gamma \in (0,1)$, $\Gamma_1 > 0, \Gamma_2 > 0$, $\sigma \geq 0$.

For $k = 0, 1, 2, \ldots$

Set $L_{\texttt{trial}} = L_k$

For all $\boldsymbol{x}_c \in L_k$ **and if** $\boldsymbol{x}_c \in L_{\texttt{trial}}$ **do**

Phase 1: Set $I = \{1, \ldots, m\}$. If $\theta_I(\boldsymbol{x}_c) < -\sigma$, then compute a common descent direction as defined in (3.8). If $\max\limits_{\ell=1,\ldots,m} \nabla f_\ell(\boldsymbol{x}_c)^\top v_D^I(\boldsymbol{x}_c) \leq -\Gamma_1 \|v^I(\boldsymbol{x}_c)\|^2$ and $\|v_D^I(\boldsymbol{x}_c)\| \leq \Gamma_2 \|v^I(\boldsymbol{x}_c)\|^2$, then set $\boldsymbol{d}_c = v_D^I(\boldsymbol{x}_c)$; if not, $\boldsymbol{d}_c = v^I(\boldsymbol{x}_c)$. Compute $\alpha_c^k = \max\limits_{h \in \mathbb{N}} \{\alpha_0 \delta^h \mid F(\boldsymbol{x}_c + \alpha_0 \delta^h d_c) \leq F(\boldsymbol{x}_c) + \mathbf{1}\gamma\alpha_0\delta^h \max_{\ell \in I} \nabla f_\ell(\boldsymbol{x}_c)^\top \boldsymbol{d}_c\}$. Otherwise, set $\boldsymbol{d}_c = v^I(\boldsymbol{x}_c)$, $\alpha_c^k = 0$. Set $z_c^k = \boldsymbol{x}_c + \alpha_c^k d_c$ and call $L_{\texttt{filtered}} = \texttt{filter}(z_c^k, L_{\texttt{trial}})$ to remove all dominated points from $L_{\texttt{trial}} \cup \{z_c^k\}$. Set $L_{\texttt{trial}} = L_{\texttt{filtered}}$.

Phase 2:

For all $I' \subset \{1, \ldots, m\}$ s.t. $\theta_{I'}(\boldsymbol{z}_c^k) < 0$, and $\boldsymbol{z}_c^k \in L_{\texttt{trial}}$ **do**
Compute a partial descent direction I' as defined in (3.9), and $\alpha_c^{I'} = \max\limits_{h \in \mathbb{N}} \{\alpha_0 \delta^h \mid \forall \boldsymbol{y} \in L_{\texttt{trial}},\ \exists\, \ell \text{ s.t. } f_\ell(\boldsymbol{z}_c^k + \alpha_0 \delta^h v^{I'}(\boldsymbol{z}_c^k)) < f_\ell(\boldsymbol{y})\}$. Call $L_{\texttt{filtered}} = \texttt{filter}(\boldsymbol{z}_c^k + \alpha_c^{I'} v^{I'}(\boldsymbol{z}_c^k), L_{\texttt{trial}})$ to remove all dominated points from $L_{\texttt{trial}} \cup \{\boldsymbol{z}_c^k + \alpha_c^{I'} v^{I'}(\boldsymbol{z}_c^k)\}$. Set $L_{\texttt{trial}} = L_{\texttt{filtered}}$.
EndFor

EndFor

Set $L_{k+1} = L_{\texttt{trial}}$.

EndFor

the sequence of sets of nondominated points produced by Algorithm 4 under Assumption 3.2.2. Then,

(i) if $\sigma > 0$, there exists $\bar{k}$ such that $\Theta(L_k) \geq -\sigma$ for all $k \geq \bar{k}$;

(ii) if $\sigma = 0$, $\lim\limits_{k \to +\infty} \Theta(L_k) = 0$.

Trust-region methods have also been adapted to address multiobjective derivative-based optimization problems. Similar to scalar optimization, each iteration involves solving a subproblem and decrease is achieved by adapting the concepts of actual and predicted reductions to the vectorial context.

In [168] a trust-region algorithm was introduced, aiming at finding stationary points in unconstrained multiobjective optimization problems, assuming that the models Hessians as positive definite. In [42], a trust-region-based algorithm for nonconvex unconstrained multiobjective optimization problems was proposed, as a generalization of the method developed in [89] for convex problems. Under differentiability assumptions, the algorithm generates a sequence of points that converges to a point satisfying a necessary condition for

weak Pareto optimality and, in the convex case, to a Pareto point, meeting both necessary and sufficient conditions. In [170], the authors proposed a trust-region algorithm based on a nonmonotone technique for solving unconstrained multiobjective optimization problems, adapting the method proposed in [42].

However, all the algorithms discussed in these papers [42, 168, 170] are designed to find a single stationary point of the problem. More recently, [146] proposed an algorithm based on a trust-region approach to approximate the set of Pareto critical points of a multiobjective optimization problem. Assuming known derivatives, this method computes quadratic Taylor models for the different objective function components, which are minimized in two main steps: the extreme point step and the scalarization step, the first aiming at reach the extreme points and the latter at filling the existing gaps.

Fliege and Vaz proposed in [91] a method for constrained and unconstrained nonlinear multiobjective optimization problems based on a sequential quadratic programming (SQP) approach. This approach uses the derivative information of the objective functions and constraints (both inequality and equality constraints) to construct a quadratic approximation of all the objective function components and a linear approximation of all the constraints, from which search directions are derived. The algorithm maintains a list of nondominated points at each iteration with the aim of approximating the Pareto front. Under appropriate differentiability assumptions, convergence to Pareto critical points is proven.

Recently, in [6], similarly to the method proposed in [91], an SQP with a new line-search technique was developed for nonlinear multiobjective optimization problems. Differently from [91], this method uses the derivative information to build a quadratic approximations both for the objective function components and the constraints, from which search directions are derived.

An adaptation of the classical augmented Lagrangian method to handle inequality constrained multiobjective problems, namely ALAMO, was proposed in [48]. Under differentiability assumptions and convexity of the constraints, the algorithm generates a sequence of points that converges to a Pareto stationary point. However, this method generates only one Pareto stationary point and does not capture a good Pareto front approximation. Later, in [49], an extension of ALAMO was proposed to generate a sequence of sets of nondominated points and produce good approximations of the entire Pareto front, namely Front-ALAMO.

As we saw, when solving a multiobjective optimization problem, the goal is to identify a set of nondominated points. These points are characterized by the fact that it is impossible to simultaneously improve all components of the objective function. This task is even harder if derivatives are not available [14, 60]. In this case, the problem to be solved would be a multiobjective derivative-free optimization problem. Like in single-objective DFO, multiobjective optimization also encompasses three main classes of methods: trust-region, direct search, and unidirectional search methods.

Trust-region methods based on quadratic polynomial interpolation models have been

proposed for unconstrained biobjective DFO problems [174]. These algorithms take advantage of the Taylor-like bounds that can be established for the errors between these models and the true function, proceeding by minimizing each model by itself and a joint combination of the two models, using a scalarization approach [80]. Resulting dominated points are not accepted and the procedure is repeated for the most isolated point. This work was extended to general bound constrained multiobjective optimization problems [74], in this last case without rigorously establishing the corresponding convergence analysis. In [145], the approach developed in [146] was adapted to solve multiobjective DFO problems by simply replacing the Taylor-based quadratic models with models computed with interpolation or minimum Frobenius norm approaches.

The first algorithm of directional direct search type developed for multiobjective optimization was BiMADS [18], restricted to the biobjective case. Essentially, it generates a sequence of single-objective reformulations of the biobjective optimization problem and solves them sequentially using MADS. The algorithm exploits the property that Pareto points may be ordered in biobjective problems, but not in general multiobjective ones. The Pareto front approximation is shown to satisfy some first-order necessary optimality conditions for nonsmooth functions.

To extend BiMADS to general multiobjective optimization, [17] proposed MULTI-MADS. Like BiMADS, it generates an approximation of the Pareto front by using MADS to solve a sequence of single-objective reformulations of the multiobjective optimization problem, created using the NBI [68] framework. Again, the Pareto front approximation is shown to satisfy some first-order necessary optimality conditions for nonsmooth functions. However, the corresponding implementation was never available to the community.

The first practical algorithm capable of generating an approximation of the complete Pareto front of a general multiobjective DFO problem, with stated theoretical convergence, was Direct Multisearch (DMS), which generalizes directional direct search to multiobjective optimization [64]. As a directional direct search method, each iteration of DMS follows the search/poll paradigm, with the search step being optional, as the convergence results from the poll step of the algorithm. Polynomial interpolation models were incorporated into the search step to enhance the method's numerical performance [37]. This class of algorithms has proven to be competitive in both academic problems and real-world applications, even when compared to derivative-based multiobjective optimization algorithms [5]. A description of the DMS framework is provided in Algorithm 5.

Algorithm 6 is used for filtering the dominated points, i.e., for removing dominated points from the union $L_1 \cup L_2$, ensuring no duplicate elements, given that L_1 consists exclusively of nondominated points.

The convergence of DMS has been established, closely following the arguments used in the analysis of single-objective directional direct search methods, as stated for Algorithm 3. Therefore, after guaranteeing the existence of a subsequence of stepsize parameters converging to zero, this property is used for establishing that there is no locally improving feasible direction.

Algorithm 5 Direct Multisearch for MOO

Initialization

Choose $\boldsymbol{x}_0 \in \Omega$ such that $f_\ell(\boldsymbol{x}_0) < +\infty$, $\forall \ell \in \{1, \ldots, m\}$, $\alpha_0 > 0$ an initial stepsize parameter, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, with directions $\boldsymbol{d}$ satisfying $0 < d_{\min} \leq \|\boldsymbol{d}\| \leq d_{\max}$. Consider $L_0 = \{(\boldsymbol{x}_0; \alpha_0)\}$ the initial list of nondominated points and corresponding stepsize parameters.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according to some criteria and select $(\boldsymbol{x}_k; \alpha_k) \in L_k$ as the current iterate and stepsize parameter.

2. **Search step:** Compute a finite set of points $\{\boldsymbol{z}_s\}_{s \in S}$ (in a mesh if $\bar{\rho}(\cdot) \equiv 0$). Evaluate F_Ω (as defined in (3.4)) at each element of $\{\boldsymbol{z}_s\}_{s \in S}$. Set $L_{\texttt{add}} = \{(\boldsymbol{z}_s; \alpha_k), s \in S\}$. Call $L_{\texttt{trial}} = \texttt{filter}(L_k, L_{\texttt{add}})$ to eliminate dominated points from $L_k \cup L_{\texttt{add}}$ (see Algorithm 6). If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the search step) successful, set $L_{k+1} = L_{\texttt{trial}}$, and go to Step 4.

3. **Poll step:** Choose a positive spanning set $\boldsymbol{D}_k$ from the set $\mathcal{D}$. Evaluate F_Ω at $P_k = \{\boldsymbol{x}_k + \alpha_k \boldsymbol{d} \mid \boldsymbol{d} \in \boldsymbol{D}_k\}$. Set $L_{\texttt{add}} = \{(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k), \boldsymbol{d} \in \boldsymbol{D}_k\}$. Call $L_{\texttt{trial}} = \texttt{filter}(L_k, L_{\texttt{add}})$ to eliminate dominated points from $L_k \cup L_{\texttt{add}}$ (see Algorithm 6). If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the poll step) as successful and set $L_{k+1} = L_{\texttt{trial}}$. Otherwise, declare the iteration as unsuccessful and set $L_{k+1} = L_k$.

4. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma \alpha_k]$. Replace all the new points $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_{k,\texttt{new}})$, when success is coming from the poll step, or $(\boldsymbol{z}_s; \alpha_k)$ in L_{k+1} by $(\boldsymbol{z}_s; \alpha_{k,\texttt{new}})$, when success is coming from the search step. Replace also $(\boldsymbol{x}_k; \alpha_k)$, if in L_{k+1}, by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$. Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,\texttt{new}} \in [\beta_1 \alpha_k, \beta_2 \alpha_k]$, and replace the poll pair $(\boldsymbol{x}_k; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$.

EndFor

Theorem 3.2.4. *[64, Thm. 4.8] Consider a refining subsequence $\{\boldsymbol{x}_k\}_{k \in K}$ generated by Algorithm 5 converging to $\bar{\boldsymbol{x}} \in \Omega$ and a refining direction $\boldsymbol{d}$ for $\bar{\boldsymbol{x}}$ in $H_\Omega^{Cl}(\bar{\boldsymbol{x}})$. Assume that $\boldsymbol{F}$ is Lipschitz continuous near $\bar{\boldsymbol{x}}$. Then, there exists a $\ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\}$ such that $f_\ell^\circ(\bar{\boldsymbol{x}}; \boldsymbol{d}) \geq 0$. Moreover, if the function $\boldsymbol{F}$ is strictly differentiable at $\bar{\boldsymbol{x}}$, then there exists a $\ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\}$ such that $\nabla f_\ell(\bar{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0$.*

Under an asymptotic density assumption of the set of refining directions in the Clarke tangent cone to Ω computed at limit points of refining subsequences, the convergence to Pareto-Clarke critical points or Pareto-Clarke-KKT critical points is established.

Theorem 3.2.5. *[64, Thm. 4.9] Consider a refining subsequence $\{\boldsymbol{x}_k\}_{k \in K}$ converging to $\bar{\boldsymbol{x}} \in \Omega$. Assume that $\boldsymbol{F}$ is Lipschitz continuous near $\bar{\boldsymbol{x}}$ and $H_\Omega^{Cl}(\bar{\boldsymbol{x}}) \neq \emptyset$. If the set of refining directions for*

Algorithm 6 Filtering procedure

$[L_3] = \texttt{filter}(L_1, L_2)$
Set $L_3 = L_1 \cup L_2$

for all $x \in L_2$

do
- **for all** $y \in L_3, y \neq x$ **do**
  - **if** $y \prec x$ **then** $\{L_3 = L_3 \setminus \{x\}\}$
- **if** $x \in L_3$ **then**
  - **for all** $y \in L_3, y \neq x$ **do**
    - **if** $x \prec y$ **then** $L_3 = L_3 \setminus \{y\}$
    - **if** $y \in L_2$ **then** $L_2 = L_2 \setminus \{y\}$

$\bar{\boldsymbol{x}}$ is dense in $T_\Omega^{Cl}(\bar{\boldsymbol{x}})$, then $\bar{\boldsymbol{x}}$ is a Pareto-Clarke critical point, i.e,

$$\forall \boldsymbol{d} \in T_\Omega^{Cl}(\bar{\boldsymbol{x}}), \exists \ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\} \text{ such that } f_\ell^\circ(\bar{\boldsymbol{x}}; \boldsymbol{d}) \geq 0.$$

If, in addition, $\boldsymbol{F}$ is strictly differentiable at $\bar{\boldsymbol{x}}$, then $\bar{\boldsymbol{x}}$ is a Pareto-Clarke-KKT critical point, i.e.,

$$\forall \boldsymbol{d} \in T_\Omega^{Cl}(\bar{\boldsymbol{x}}), \exists \ell = \ell(\boldsymbol{d}) \in \{1, \ldots, m\} \text{ such that } \nabla f_\ell(\bar{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0.$$

Considering a sufficient decrease approach for accepting new points, worst-case complexity bounds were provided for DMS [63]. Moreover, for a particular algorithmic instance, which considers a stricter criterion for accepting new nondominated points, DMS presents a worst-case complexity bound of $O(\epsilon^{-2})$, which is similar to the ones of steepest descent or Newton's methods.

Inspired by the works [17, 18, 64], Bigeon *et al.* [28] proposed a new extension of the MADS algorithm for bound-constrained multiobjective optimization. Unlike the previous BiMADS and MULTIMADS methods, the DMultiMADS algorithm does not solve a sequence of single-objective parameterized reformulations. Instead, it maintains a list of nondominated points at each iteration, as in DMS.

This new extension of MADS for multiobjective optimization shares similarities with DMS but differs in how it handles the choice of incumbent points during each iteration. As we saw, DMS generates at least one refining subsequence in which the refined point converges to a Pareto-Clarke critical point. In DMultiMADS, every linked sequence generated by the algorithm is a refining subsequence. However, this does not imply that the algorithm generates more than one linked sequence.

Later, in [29], two new constraint-handling strategies were proposed for DMultiMADS [28]. The constraints are aggregated into a single constraint violation function which is used either in a two-phases approach (DMultiMADS-TEB) or in a progressive

barrier approach (DMultiMADS-PB). In DMultiMADS-TEB, the constraint violation function is minimized with the MADS [13] algorithm, in a single-objective setting, until a feasible point is found. After, the algorithm proceeds as in DMultiMADS [28], making use of an extreme barrier function, only evaluating feasible points. The second approach (DMultiMADS-PB) generalizes progressive barrier [11] to multiobjective optimization. The constraint violation function is considered as an additional objective to be minimized, being rejected any trial point with constraint violation function value above a given threshold. This threshold is progressively decreased along the iterations. Each iteration explores two poll centers, corresponding to feasible and infeasible points, respectively. Progressive barrier [11] can be regarded as an evolution of filter methods [10], initially proposed by Fletcher and Leyffer [87] for single-objective derivative-based optimization, as an alternative to address general constraints.

In [134], the authors considered a black-box general nonlinear constrained multiobjective optimization problem and proposed algorithm DFMO for its solution. The DFMO algorithm is based on a new exact-penalty derivative-free approach that utilizes line-searches with sufficient decrease. An equivalence between the original problem and a bound-constrained problem obtained by penalizing the nonlinear constraints with an exact merit function was established. Moreover, under certain density assumptions on the search directions, Pareto criticality was established for any accumulation point of the sequences generated by the method.

Finally, the unidirectional class, comprises generalization of implicit filter [120] for bound constrained multiobjective derivative-free optimization [51]. At each iteration of the proposed algorithm, a simplex Jacobian is defined and used to compute an approximation to the multiobjective steepest descent direction, which is then explored in a line-search. Under smooth assumption, convergence is stated.

We can also have the situation where some components of the objective function are expensive black-box functions, such as a time-consuming simulations without derivative information, while others are analytically given, with easily computable derivatives. In [180], a trust-region approach was proposed to address these heterogeneous problems. Quadratic Taylor models are used to approximate the inexpensive function components and quadratic interpolation based on Lagrangian polynomials is considered in the remaining situations. In its simplest variant, the algorithm generates a single sequence of points, whose limit points are proved to be Pareto critical.

3.2.1 Performance assessment in multiobjective optimization

With adaptations, performance profiles, already described in Subsection 3.1.1, can also be used to evaluate the numerical performance of multiobjective solvers. Recall that the performance of a solver s belonging to the set S on a set of problems P is represented

using the cumulative function:

$$\rho_s(\tau) = \frac{1}{|P|} \left| \{ p \in P \mid t_{p,s} \leq \tau \min\{t_{p,s} : s \in S\} \} \right|$$

where $\tau \geq 1$ and $t_{p,s}$ represents the value of the selected metric, obtained by solver $s \in S$ when solving problem $p \in P$. Implicitly, it is assumed that $t_{p,s} \geq 0$ and the lowest, the better.

It is also worth recalling that higher values of $\rho_s(\tau)$ indicate superior numerical performance for solver s. Specifically, the solver with the highest $\rho_s(1)$ value is considered the most efficient. Conversely, the solver with the greatest $\rho_s(\tau)$ value for large τ values is regarded as the most robust.

As metrics, purity [21], hypervolume [189], and the spread metrics Γ and Δ [64] are commonly used in the multiobjective literature. Purity measures the percentage of nondominated points generated by a given solver, corresponding to the ratio:

$$\bar{t}_{p,s} = Pur_{p,s} = \frac{|F_{p,s} \cap F_p|}{|F_{p,s}|},$$

where $F_{p,s}$ denotes the Pareto front approximation computed by solver s for problem p and F_p represents a reference Pareto front for problem p, obtained by aggregating the Pareto approximations from all solvers, $\cup_{s \in S} F_{p,s}$, and eliminating all dominated points from it [64].

Hypervolume, proposed in [189], additionally to nondominance seeks to encompass the notion of spread, by quantifying the volume of the region dominated by the current approximation of the Pareto front in relation to a reference point $\boldsymbol{u}_p \in \mathbb{R}^m$,

$$\bar{t}_{p,s} = HV_{p,s} = \mathcal{V}\{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{y} \leq \boldsymbol{u}_p \wedge \exists \boldsymbol{x} \in F_{p,s} : \boldsymbol{x} \leq \boldsymbol{y}\} = \mathcal{V}\left(\bigcup_{\boldsymbol{x} \in F_{p,s}} [\boldsymbol{x}, \boldsymbol{u}_p] \right), \tag{3.10}$$

where $\mathcal{V}(\cdot)$ denotes the Lebesgue measure of a m-dimensional set of points and $[\boldsymbol{x}, \boldsymbol{u}_p]$ denotes the interval box with lower corner $\boldsymbol{x}$ and upper corner $\boldsymbol{u}_p$.

For calculating the performance profiles for purity and hypervolume metrics, since larger values indicate superior performance, the inverse of each metric ($t_{p,s} = \frac{1}{\bar{t}_{p,s}}$) is used.

Finally, to provide a comprehensive assessment of spread across the Pareto front, two additional metrics, Γ and Δ, are considered. The Γ metric measures the size of the largest gap in the Pareto front approximation, while the Δ metric evaluates the uniformity of the distribution of the nondominated points within the generated approximation.

Consider that for problem $p \in P$, solver $s \in S$ has computed an approximation to the Pareto front consisting of points $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N$. To this set, we add the extreme points, denoted as $\boldsymbol{x}^0$ and $\boldsymbol{x}^{N+1}$, which represent the points with the best and worst values for component ℓ of the objective function, respectively, computed taking into consideration all the final approximations to the Pareto front of problem p computed by all solvers $s \in S$.

The metric Γ can be computed as:

$$\Gamma_{p,s} = \max_{\ell \in \{1,\ldots,m\}} \left(\max_{i \in \{0,\ldots,N\}} \{\delta_{\ell,i}\} \right), \tag{3.11}$$

where $\delta_{\ell,i} = f_\ell(\boldsymbol{x}^{i+1}) - f_\ell(\boldsymbol{x}^i)$, assuming that the objective function values have been arranged in ascending order for each component $\ell \in \{1,\ldots,m\}$ of the objective function. The metric Δ is computed as:

$$\Delta_{p,s} = \max_{\ell \in \{1,\ldots,m\}} \left(\frac{\delta_{\ell,0} + \delta_{\ell,N} + \sum_{i=1}^{N-1} |\delta_{\ell,i} - \bar{\delta}_\ell|}{\delta_{\ell,0} + \delta_{\ell,N} + (N-1)\bar{\delta}_\ell} \right), \tag{3.12}$$

where $\bar{\delta}_\ell$, for $\ell = 1,\ldots,m$, represents the average of the distances $\delta_{\ell,i}$, $i = 1,\ldots,N-1$ (see [72]).

3.3 Gaps in the existing literature

Multiobjective derivative-free optimization problems are often addressed through heuristics, such as evolutionary or genetic algorithms [36]. However, if the cost of computing a function value is high, these methods may not be appropriate, as they often require a large number of function evaluations. In the last decade, specific classes of derivative-free optimization methods have been developed for multiobjective optimization. In fact, a recent survey [126] considers multiobjective derivative-free optimization to be "an especially open avenue of future research".

As we noted, in single-objective derivative-free optimization, there are three main classes of methods [60], which have generalizations proposed for multiobjective optimization. The first, directional direct search, has been generalized to multiobjective optimization with Direct Multisearch [64]. Trust-region methods based on quadratic polynomial interpolation models have been proposed for unconstrained biobjective derivative-free optimization problems [174] and extended to general bound-constrained multiobjective optimization problems [145]. The algorithm proposed in [51] generalizes implicit filtering, a unidirectional search method, to bound-constrained derivative-free multiobjective optimization. For all three mentioned algorithmic classes, convergence is established and numerical results support the ability to compute good approximations to the complete Pareto front of a given problem. However, constraints have not been fully addressed.

Despite the growing interest in multiobjective derivative-free optimization, there remains a notable lack of literature addressing constrained scenarios. Most existing works focus on bound-constrained problems, with limited exploration of general nonlinear constraints. The works [18, 51, 74, 174] only tackle bound-constrained multiobjective optimization problems. From a theoretical perspective, Direct Multisearch [64] is developed for continuous optimization with general constraints; however, it adopts an extreme barrier approach, evaluating only feasible points. In [134], an exact penalty approach to general

nonlinear constraints was considered. Additionally, [29] proposed two new constraint-handling strategies for DMultiMADS [28]: a two-phase approach (DMultiMADS-TEB) and a progressive barrier approach (DMultiMADS-PB), which, to our knowledge, are the only works that explicitly address general constraints in multiobjective derivative-free optimization.

This gap highlights a significant opportunity for further research to develop comprehensive frameworks that can effectively handle a wide variety of constraints in multiobjective settings, thereby enhancing the applicability and robustness of multiobjective derivative-free optimization methods in real-world problems. Thus, the main goal of this thesis is to develop strategies to explicitly address general constraints in multiobjective derivative-free optimization. The corresponding value will be analyzed from a theoretical standpoint and a numerical assessment will also be conducted.

4

A Filter-Based Direct Multisearch Approach with Inexact Restoration for Constrained Optimization

In this chapter, we address the multiobjective optimization problem with general constraints (linear and nonlinear), defined by:

$$\begin{aligned} \min_{x \in X} \quad & \mathbf{F}(x) = (f_1(x), \ldots, f_m(x))^\top \\ \text{s.t.} \quad & \mathbf{g}(x) \leq \mathbf{0}, \end{aligned} \tag{4.1}$$

where $\mathbf{F} : X \subseteq \mathbb{R}^n \to \{\mathbb{R} \cup \{+\infty\}\}^m$, with $m \in \mathbb{N}$, $m \geq 2$, $\mathbf{g} : X \subseteq \mathbb{R}^n \to \{\mathbb{R} \cup \{+\infty\}\}^p$, $p \in \mathbb{N}$, and $X \subseteq \mathbb{R}^n$ denotes the set of unrelaxable constraints [60]. Therefore, the feasible region, Υ, of the multiobjective problem, assumed to be nonempty, is given by $\Upsilon = X \cap \Omega \neq \emptyset$, where $\Omega = \{x \in \mathbb{R}^n \mid \mathbf{g}(x) \leq \mathbf{0}\}$.

The main goal of this thesis is to develop strategies for explicitly addressing general constraints in multiobjective derivative-free optimization. Filter methods are an effective approach for it [87]. Originally proposed in derivative-based optimization, a filter method can be viewed as a biobjective approach that aims to minimize the objective function and the constraint violation simultaneously. As seen in Section 3.1, this strategy has been previously applied in single-objective directional direct search [10]. Therefore, to tackle Problem 4.1, we propose the replacement of the extreme barrier approach by a filter strategy, combined with an inexact feasibility restoration step, to address constraints in the DMS framework. The filter approach treats feasibility as an additional component of the objective function, that needs to be minimized. The inexact restoration step attempts to generate new feasible points, contributing to prioritize this feasibility, a requirement for the good performance of any filter approach.

The work presented in this chapter is based on the study published in [176]. In Section 4.1 we describe the proposed algorithmic structure. The theoretical properties of the sequences of points generated by the algorithm are established in Section 4.2. Section 4.3 provides some details respecting the numerical implementation used to compute the results reported in Section 4.4.

4.1 Algorithm structure

In applications, unrelaxable constraints are often associated to physical conditions that can not be violated (otherwise, it will be impossible to evaluate the objective function). Thus, our approach will address them with an extreme barrier function, only evaluating points that satisfy these constraints. In the problem definition, function $\mathbf{F}$ will be replaced by $\mathbf{F}_X$, defined as:

$$\mathbf{F}_X(\boldsymbol{x}) = \begin{cases} \mathbf{F}(\boldsymbol{x}) & , \text{ if } \boldsymbol{x} \in X \\ (+\infty, +\infty, \ldots, +\infty)^\top & , \text{ otherwise.} \end{cases} \tag{4.2}$$

Although, points do not always need to remain feasible regarding the relaxable constraints, defined by function $\mathbf{g}(\cdot)$. We intend to minimize this violation and a maximum threshold, $h_{\max} > 0$, will be allowed for it. For that, following the approach of [10], proposed for single-objective optimization, we will consider an additional nonnegative objective function component, h, corresponding to an aggregated violation of the relaxable constraints. Function h should satisfy $h(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} \in \Omega$. A possibility for its definition could be:

$$h(\boldsymbol{x}) = \| \mathbf{g}(\boldsymbol{x})_+ \|^r, \tag{4.3}$$

where $\| \cdot \|$ is a vector norm, $r > 0$, and $\mathbf{g}(\boldsymbol{x})_+$ is the vector of p constraint values, defined for $i = 1, \ldots, p$ by

$$g_i(\boldsymbol{x})_+ = \begin{cases} g_i(\boldsymbol{x}) & , \text{ if } g_i(\boldsymbol{x}) > 0 \\ 0 & , \text{ otherwise.} \end{cases} \tag{4.4}$$

As an example, considering the ℓ_2-norm and $r = 2$, we have:

$$h(\boldsymbol{x}) = \| \mathbf{g}(\boldsymbol{x})_+ \|_2^2 = \sum_{i=1}^{p} \max\{0, g_i(\boldsymbol{x})\}^2.$$

An approximation to the solution of Problem (4.1) will be computed by solving

$$\min_{\boldsymbol{x} \in X} \ \bar{\mathbf{F}}_{\bar{X}}(\boldsymbol{x}), \tag{4.5}$$

where $\bar{\mathbf{F}}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}), h(\boldsymbol{x}))^\top$ and $\bar{X} = \{\boldsymbol{x} \in X \mid h(\boldsymbol{x}) \leq h_{\max}\}$.

The algorithm proposed to address Problem (4.5) is developed under the DMS framework. Thus, a list of feasible nondominated points, regarding the unrelaxable constraints, and corresponding stepsizes is kept and updated by the algorithm during the optimization process. In a simplified way, each iteration tries to improve this list of points, by adding new nondominated points to it and removing dominated ones. The procedure follows Algorithm 6 (see Section 3.2). At the end of the optimization process, the points in the list that satisfy $h(\boldsymbol{x}) = 0$ constitute the approximation to the Pareto front of the original problem.

Each iteration starts with the selection of an iterate point (and corresponding stepsize) from the list. This iterate point is always feasible regarding the constraints defining the set

X, but can be infeasible with respect to the constraints defining the set Ω. In Section 4.3, we will propose practical rules for the selection of iterate points. After this selection, following the classical structure of a directional direct search method, a search step and possibly a poll step are performed.

The search step is optional, not required for establishing convergence, allowing the definition of many strategies, with the main purpose of improving the algorithmic performance. The minimization of quadratic polynomial models has already been proposed in the DMS framework [37]. It is the poll step that is responsible for the theoretical properties of the algorithm, so we will focus on it. At this step, a local search is performed around the current iterate, by exploring directions belonging to a positive spanning set, scaled by the stepsize. Details on the properties that this set of directions needs to satisfy will be provided in Section 4.2.

Due to the presence of relaxable constraints, the iterate point could be infeasible regarding the set Ω (and the original problem). In this situation, considering the expensive nature of the objective function, it would be wiser to try to restore feasibility, before initiating the polling procedure. For that, the following inexact feasibility restoration problem will be solved:

$$\begin{aligned} &\min_{\boldsymbol{y}\in X} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}_k\|^2 \\ &\text{s.t. } h(\boldsymbol{y}) \leq \xi(\alpha_k)h(\boldsymbol{x}_k), \end{aligned}$$

where $\boldsymbol{x}_k$ and α_k denote the current iterate and the associated stepsize, respectively. Function $\xi : (0, +\infty) \to (0, 1)$ is continuous and satisfies $\xi(t) \to 0$ when $t \downarrow 0$. Since the feasible region Υ is nonempty, assuming, for example, X as a compact set, this problem is well-defined. Solving the inexact restoration problem, before polling being attempted, is an explicit way of prioritizing feasibility. The definition of the $\xi(\cdot)$ function ensures that if the stepsize goes to zero, in general meaning that a limit point is being attained, then feasibility regarding Ω is also being restored.

The list of points is a dynamic set, that will allow to classify iterations as being successful or unsuccessful. Similarly to the original implementation of DMS [64], an iteration is said to be successful if the iterate list changes, meaning that at least one new feasible nondominated point was added to it. Here feasibility respects only to the set X, of unrelaxable constraints, and dominance function $\bar{\mathbf{F}}$. Unsuccessful iterations keep the list unchanged. Differently from [29], comparisons are made for all the points in the list, regardless of the associated feasibility.

The rule for updating the stepsize parameter follows what is classical in directional direct search. Therefore, for successful iterations, the stepsize parameter is either increased or kept constant, i.e., $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma\alpha_k]$ for $\gamma \geq 1$, for all the feasible nondominated points added and for the poll center, if it remains in the list. At unsuccessful iterations, the stepsize of the poll center is decreased, i.e., $\alpha_{k,\texttt{new}} \in [\beta_1\alpha_k, \beta_2\alpha_k]$ for $0 < \beta_1 \leq \beta_2 < 1$.

Algorithm 7 details the DMS-FILTER-IR multiobjective derivative-free constrained optimization method.

Algorithm 7 DMS-FILTER-IR method for constrained MOO

Initialization

Choose an initial stepsize parameter $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, with directions $\boldsymbol{d}$ satisfying $0 < d_{min} \leq \|\boldsymbol{d}\| \leq d_{max}$. Define $h(\cdot)$, the nonnegative violation aggregation function, and $h_{\max} > 0$, the maximum violation allowed for it. Define $\xi(\cdot)$, to be used in the inexact restoration step. Consider $x_0 \in X$ such that $h(\boldsymbol{x}_0) \leq h_{\max}$ and set $L_0 = \{(\boldsymbol{x}_0; \alpha_0)\}$.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according to some criteria and select $(\boldsymbol{x}_k; \alpha_k) \in L_k$ as the current iterate and stepsize parameter.

2. **Search step:** Compute a finite set of points $\{\boldsymbol{z}_s\}_{s \in S}$ (in a mesh if $\bar{\rho}(\cdot) \equiv 0$, see Section 4.2). Evaluate $\bar{\mathbf{F}}_{\bar{X}}$ at each element of $\{\boldsymbol{z}_s\}_{s \in S}$. Use $L_{\texttt{add}} = \{(\boldsymbol{z}_s; \alpha_k), s \in S\}$ to generate $L_{\texttt{trial}}$, by updating L_k with the new nondominated points in $L_{\texttt{add}}$ and removing the dominated ones. If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the search step) successful, set $L_{k+1} = L_{\texttt{trial}}$, and go to Step 5.

3. **Inexact Restoration step:** If $h(\boldsymbol{x}_k) > 0$ then define and solve the inexact restoration problem $\boldsymbol{y}^* \in \arg\min_{\boldsymbol{y} \in X} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}_k\|^2$ subject to $h(\boldsymbol{y}) \leq \xi(\alpha_k) h(\boldsymbol{x}_k)$ (in a mesh if $\bar{\rho}(\cdot) \equiv 0$, see Section 4.2). Evaluate $\bar{\mathbf{F}}_{\bar{X}}$ at $\boldsymbol{y}^*$, define $L_{\texttt{add}} = \{(\boldsymbol{y}^*; \alpha_k)\}$ to generate $L_{\texttt{trial}}$, by updating L_k with the new nondominated point in $L_{\texttt{add}}$ and removing the dominated ones. If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the inexact restoration step) as successful, set $L_{k+1} = L_{\texttt{trial}}$, and go to Step 5.

4. **Poll step:** Choose a positive spanning set $\boldsymbol{D}_k$ from the set $\mathcal{D}$. Evaluate $\bar{\mathbf{F}}_{\bar{X}}$ at $P_k = \{\boldsymbol{x}_k + \alpha_k \boldsymbol{d} \mid \boldsymbol{d} \in \boldsymbol{D}_k\}$, define $L_{\texttt{add}} = \{(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k), \boldsymbol{d} \in \boldsymbol{D}_k\}$ to generate $L_{\texttt{trial}}$, by updating L_k with the new nondominated points in $L_{\texttt{add}}$ and removing the dominated ones. If $L_{\texttt{trial}} \neq L_k$, then declare the iteration (and the poll step) as successful and set $L_{k+1} = L_{\texttt{trial}}$. Otherwise, declare the iteration as unsuccessful and set $L_{k+1} = L_k$.

5. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma\alpha_k]$. Replace all the new points $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_{k,\texttt{new}})$, when success is coming from the poll step, or $(\boldsymbol{y}^*; \alpha_k)$ in L_{k+1} by $(\boldsymbol{y}^*; \alpha_{k,\texttt{new}})$, when success is coming from the inexact restoration step, or $(\boldsymbol{z}_s; \alpha_k)$ in L_{k+1} by $(\boldsymbol{z}_s; \alpha_{k,\texttt{new}})$, when success is coming from the search step. Replace also $(\boldsymbol{x}_k; \alpha_k)$, if in L_{k+1}, by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$. Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,\texttt{new}} \in [\beta_1\alpha_k, \beta_2\alpha_k]$, and replace the poll pair $(\boldsymbol{x}_k; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$.

EndFor

4.2 Convergence analysis

Following the reasoning of [10, 64], this section analyzes the properties of the different sequences of points generated by DMS-FILTER-IR.

4.2.1 Globalization strategies

In classical directional direct search, the first step in the convergence analysis is globalization, i.e., to ensure the existence of a subsequence of stepsize parameters that converges to zero. Two different strategies can be adopted. The first, analyzed next, requires that all points generated by the algorithm lie in an implicit mesh, corresponding to an integer lattice.

For establishing the result, we will need the following assumption.

Assumption 4.2.1. *The set* $\mathcal{S} := \bigcup_{\ell=1}^{m} \{\boldsymbol{x} \in X \mid h(\boldsymbol{x}) \leq h_{\texttt{max}} \wedge f_\ell(\boldsymbol{x}) \leq f_\ell(\boldsymbol{x}_0)\}$ *is a nonempty compact set.*

In [64], when DMS was originally proposed, the directions used by the algorithm at each iteration belonged to a positive spanning set, $\boldsymbol{D}_k$, selected from a predefined set $\mathcal{D}$. The directions in $\mathcal{D}$ were constructed as nonnegative integer combinations of the columns of a set $\boldsymbol{D}$. Assumption 3.1.4 formalizes the conditions imposed on the set $\boldsymbol{D}$ to ensure that the integrality requirements are satisfied.

Moreover, in the presence of general constraints and possibly nonsmooth functions, it is necessary to consider an infinite set of directions, $\mathcal{D}$, which must be dense (after normalization) in the unit sphere, as stated in Assumption 3.1.5. Furthermore, as in the original DMS, to comply with the integrality requirements outlined in Assumption 3.1.6, the update rule in Algorithm 7 is defined as $\beta_1 = \tau^{m^{\min}}$, $\beta_2 = \tau^{-1}$, and $\gamma = \tau^{m^{\max}}$.

In addition, the points generated both at the search and at the inexact restoration step need to lie in the implicit mesh considered at each iteration by the algorithm.

Assumption 4.2.2. *At iteration k, the search and the inexact restoration steps in Algorithm 7 only evaluate points in*

$$M_k = \bigcup_{\boldsymbol{x} \in E_k} \{\boldsymbol{x} + \alpha_k D \boldsymbol{z} \mid \boldsymbol{z} \in \mathbb{N}_0^{|D|}\},$$

where $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$ and E_k represents the set of all points evaluated by the algorithm previously to iteration k.

The following theorem states that there is at least one subsequence of iterations for which the stepsize parameter converges to zero. The proof is omitted since it uses exactly the same arguments of Theorem A.1 in [64].

Theorem 4.2.3. *Let Assumption 4.2.1 hold. Under one of the Assumptions 3.1.4 or 3.1.5 combined with Assumptions 3.1.6 and 4.2.2, DMS-FILTER-IR generates a sequence of iterates satisfying*

$$\liminf_{k \to +\infty} \alpha_k = 0.$$

Globalization can also be ensured by requiring sufficient decrease to accept new points, by means of a forcing function. A forcing function $\rho : (0, +\infty) \to (0, +\infty)$ is a continuous and nondecreasing function, that satisfies $\rho(t)/t \to 0$ when $t \downarrow 0$ (see [121]). Typical examples of forcing functions are $\rho(t) = \eta_1 t^{1+\eta_2}$, for $\eta_1, \eta_2 > 0$. Definition 4.2.4 traduces the new dominance relationship considered.

Definition 4.2.4. *Let $\boldsymbol{y}$ belong to $\bar{X}$ and L be a list of nondominated points in $\bar{X}$. We say that $\boldsymbol{y}$ is dominated if:*

$$\exists \boldsymbol{x} \in L : \ \bar{\mathbf{F}}(\boldsymbol{x}) - \rho(\alpha) \leq \ \bar{\mathbf{F}}(\boldsymbol{y}),$$

where $\rho(\cdot)$ denotes a forcing function and α the stepsize associated to the current iteration.

Figure 4.1 illustrates the situation for the list of infeasible points, L, whose images by function $\bar{\mathbf{F}}$ correspond to the green dots. $D(L) \subset \mathbb{R}^{m+1}$ represents the image of the set of points dominated by the points in L (see equation (2.23)) and $D(L; \rho(\alpha))$ denotes the set of points whose distance in the ℓ_∞ norm to $D(L)$ is no larger than $\rho(\alpha) > 0$. Points will be accepted if their image by $\bar{\mathbf{F}}$ does not belong to $D(L; \rho(\alpha))$, ensuring an increase in the hypervolume [189] associated to the list of points of at least $(\rho(\alpha))^{m+1}$ (see Lemma 3.1 in [63]).

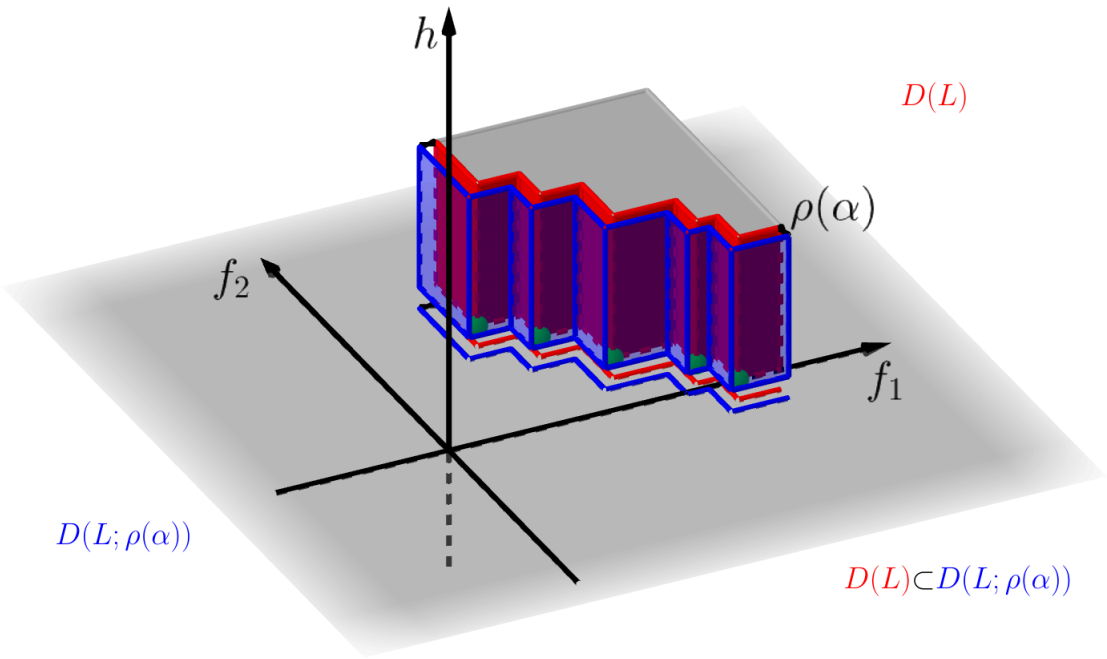


Figure 4.1: Globalization strategy based on a sufficient decrease condition in Algorithm 7.

The assumptions required to ensure globalization under a sufficient decrease approach are slightly different.

Assumption 4.2.5. *The function $F : \mathbb{R}^n \to \mathbb{R}^m$ is bounded in the set $\mathcal{S} := \bigcup_{\ell=1}^{m} \{\boldsymbol{x} \in X \mid h(\boldsymbol{x}) \leq h_{max} \wedge f_\ell(\boldsymbol{x}) \leq f_\ell(\boldsymbol{x}_0)\}$.*

By definition, $0 \leq h(\boldsymbol{x}) \leq h_{\max}$. Thus, Assumption 4.2.5 guarantees that the function $\bar{\mathbf{F}} : \mathbb{R}^n \to \mathbb{R}^{m+1}$, defined by $\bar{\mathbf{F}} = (\mathbf{F}, h)$, is also bounded in $\mathcal{S}$.

The following theorem states the convergence to zero of at least one subsequence of stepsizes, when using a globalization strategy based on sufficient decrease.

Theorem 4.2.6. *Consider a globalization strategy based on imposing a sufficient decrease condition and let Assumption 4.2.5 hold. Then, DMS-FILTER-IR generates a sequence of iterates satisfying*

$$\liminf_{k \to +\infty} \alpha_k = 0.$$

Proof. Let us assume that $\liminf_{k \to +\infty} \alpha_k \neq 0$, meaning that there is α^* such that $\alpha_k > \alpha^*$ for all k. At each unsuccessful iteration k, the corresponding stepsize parameter is reduced by at least $\beta_2 \in (0, 1)$. Thus, the number of successful iterations must be infinite.

Successful iterations increase the hypervolume of the dominated region associated to the function $\bar{\mathbf{F}}$ and the list of points in at least $(\rho(\alpha_k))^{m+1}$, where α_k represents the stepsize associated with the current iteration (see Lemma 3.1 in [63]).

Since $\rho(\cdot)$ is a nondecreasing function, which satisfies $\rho(t) > 0$ for $t > 0$, there exists $\rho^* > 0$ such that $\rho(\alpha_k) \geq \rho(\alpha^*) = \rho^*$. Thus, any successful iteration will increase the hypervolume of the dominated region associated to the list of points for function $\bar{\mathbf{F}}$ in at least $(\rho^*)^{m+1}$, contradicting Assumption 4.2.5. □

4.2.2 DMS-FILTER-IR convergence results

In this section, we present the main convergence results of DMS-FILTER-IR. For this purpose, the behavior of the algorithm needs to be analyzed at limit points of particular sequences of unsuccessful iterates, denoted by refining subsequences (see Definition 3.1.9).

Assumption 4.2.1, Theorems 4.2.3 or 4.2.6, and the updating strategy of the stepsize allow to establish the existence of at least one convergent refining subsequence (see, e.g., [60, Section 7.3]). As suggested by the numerical experiments reported in Section 4.4, it is common that the algorithm will generate several refined points. In [28] the same type of result is established using the concept of linked sequences.

The analysis is based on the property established by Proposition 4.2.7, valid for any of the two globalization strategies considered. Function $\bar{\rho}(\cdot)$ corresponds to the forcing function $\rho(\cdot)$, when globalization is based on a sufficient decrease condition, or is defined as the null function ($\bar{\rho}(\cdot) \equiv 0$), when globalization results from the use of integer lattices. For simplicity when stating the results, we also define f_{m+1} as being equal to h, the aggregated violation of the relaxed constraints.

Proposition 4.2.7. *Let $\boldsymbol{x} \in L$ and $\boldsymbol{y} \in \bar{X}$ be a dominated point at an iteration associated with the stepsize α. Then:*

$$\exists \ell \in \{1, \ldots, m+1\} : f_\ell(\boldsymbol{y}) > f_\ell(\boldsymbol{x}) - \bar{\rho}(\alpha).$$

Proof. Point $\boldsymbol{y}$ is dominated, meaning that there is $\boldsymbol{z} \in L$ such that

$$\bar{\mathbf{F}}(\boldsymbol{z}) - \bar{\rho}(\alpha) \leq \bar{\mathbf{F}}(\boldsymbol{y}),$$

and, if $\bar{\rho}(\cdot) = 0$, $\bar{\mathbf{F}}(z) \neq \bar{\mathbf{F}}(y)$.

Suppose $\bar{\mathbf{F}}(x) - \bar{\rho}(\alpha) \geq \bar{\mathbf{F}}(y)$. Then,

$$\bar{\mathbf{F}}(x) \geq \bar{\mathbf{F}}(x) - \bar{\rho}(\alpha) \geq \bar{\mathbf{F}}(z) - \bar{\rho}(\alpha).$$

Moreover, if $\bar{\rho}(\cdot) = 0$ then $\bar{\mathbf{F}}(z) \neq \bar{\mathbf{F}}(x)$. Thus, x would be dominated by z, which is not possible, since both x and z belong to the list of nondominated points L. □

As the stepsize approaches zero, the poll step will allow to recover the local sensitivities of the objective function. Together with some proper smoothness assumptions, we can establish that there is no locally improving direction, for the adequate problem. The proof follows directly from Theorem 4.8 in [64]. For completeness, we reproduce it in Theorem 4.2.8, with the due adaptations.

Theorem 4.2.8. *Consider $\{x_k\}_{k\in K}$, a refining subsequence generated by DMS-FILTER-IR, converging to the refined point $\overline{x} \in \bar{X}$. Assume that $\bar{\mathbf{F}}$ is Lipschitz continuous near $\overline{x}$. Let $d \in H_{\bar{X}}^{Cl}(\overline{x})$ be a refining direction for $\overline{x}$, associated with the refining subsequence $\{x_k\}_{k\in K}$. Then,*

$$\exists \ell \in \{1, \ldots, m+1\} : f_\ell^\circ(\overline{x}; d) \geq 0.$$

Proof. Since $\{x_k\}_{k\in K}$ is a refining subsequence converging to $\overline{x}$, we have $\lim_{k\in K} x_k = \overline{x}$, $\lim_{k\in K} \alpha_k = 0$ and $k \in K$ is the index of an unsuccessful iteration.

Consider $K^d \subseteq K$ such that $\lim_{k\in K^d} \frac{d_k}{\|d_k\|} = d \in H_{\bar{X}}^{Cl}(\overline{x})$, with d_k a poll direction used at iteration k. Let $\ell \in \{1, \ldots, m+1\}$. Then,

$$f_\ell^\circ(\overline{x}; d) = \limsup_{\substack{x\to\overline{x}, x\in\bar{X} \\ t\downarrow 0, x+td\in\bar{X}}} \frac{f_\ell(x+td) - f_\ell(x)}{t} \geq \limsup_{k\in K^d} \frac{f_\ell(x_k + \alpha_k\|d_k\| d) - f_\ell(x_k)}{\alpha_k\|d_k\|} =$$

$$= \limsup_{k\in K^d} \left(\frac{f_\ell(x_k + \alpha_k d_k) - f_\ell(x_k) + \bar{\rho}(\alpha_k)}{\alpha_k\|d_k\|} + \frac{f_\ell(x_k + \alpha_k\|d_k\| d) - f_\ell(x_k + \alpha_k d_k)}{\alpha_k\|d_k\|} - \frac{\bar{\rho}(\alpha_k)}{\alpha_k\|d_k\|} \right) \geq$$

$$\geq \limsup_{k\in K^d} \frac{f_\ell(x_k + \alpha_k d_k) - f_\ell(x_k) + \bar{\rho}(\alpha_k)}{\alpha_k\|d_k\|} + \liminf_{k\in K^d} \left(\frac{f_\ell(x_k + \alpha_k\|d_k\| d) - f_\ell(x_k + \alpha_k d_k)}{\alpha_k\|d_k\|} - \frac{\bar{\rho}(\alpha_k)}{\alpha_k\|d_k\|} \right)$$

Since each d_k is lower bounded by $d_{min} > 0$, the definition of $\bar{\rho}(\cdot)$ and the properties of the forcing function, allow to establish $\lim_{k\in K^d} \frac{\bar{\rho}(\alpha_k)}{\alpha_k\|d_k\|} = 0$.

Moreover, the Lipschitz continuity of $\bar{\mathbf{F}}$ ensures that

$$\left| \frac{f_\ell(x_k + \alpha_k\|d_k\| d) - f_\ell(x_k + \alpha_k d_k)}{\alpha_k\|d_k\|} \right| \leq L \left\| d - \frac{d_k}{\|d_k\|} \right\|,$$

where L represents the maximum of the Lipschitz constants associated to each one of the objective function components. Thus, the fact that $\lim_{k\in K^d} \frac{d_k}{\|d_k\|} = d$ ensures that

$$\lim_{k\in K^d} \frac{f_\ell(x_k + \alpha_k\|d_k\| d) - f_\ell(x_k + \alpha_k d_k)}{\alpha_k\|d_k\|} = 0.$$

We then have,

$$f_\ell^\circ(\overline{\boldsymbol{x}};\boldsymbol{d}) \geq \limsup_{k\in K^d} \frac{f_\ell(\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k)-f_\ell(\boldsymbol{x}_k)+\bar{\rho}(\alpha_k)}{\alpha_k\|\boldsymbol{d}_k\|}.$$

Now, $k \in K^d$ is an unsuccessful iteration and $\boldsymbol{x}_k \in L$. By Proposition 4.2.7,

$$\exists \ell(k) \in \{1,\ldots,m+1\} : f_{\ell(k)}(\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k) > f_{\ell(k)}(\boldsymbol{x}_k)-\bar{\rho}(\alpha_k). \tag{4.6}$$

Since the number of components of the objective function is finite, by passing to a subsequence $K^{d,\bar{\mathbf{F}}} \subseteq K^d$ that always uses the same component of $\bar{\mathbf{F}}$ we have the desired result. □

Following the classical assumptions of directional direct search [14, 60], the convergence to Pareto-Clarke or Pareto-Clarke-KKT critical points can be proven by imposing asymptotic density in the unit sphere of the set of refining directions associated with $\overline{\boldsymbol{x}}$. From a numerical point of view, it can be achieved through random generation in the unit sphere [183], or resorting to more sophisticated strategies, like is the case of LTMADS [13] or ORTHOMADS [1] implementations.

Theorem 4.2.9. *Consider $\{\boldsymbol{x}_k\}_{k\in K}$, a refining subsequence generated by DMS-FILTER-IR, converging to the refined point $\overline{\boldsymbol{x}} \in \bar{X}$. Assume that $H_{\bar{X}}^{Cl}(\overline{\boldsymbol{x}}) \neq \emptyset$ and that $\bar{\mathbf{F}}$ is Lipschitz continuous near $\overline{\boldsymbol{x}}$. If the set of refining directions for $\overline{\boldsymbol{x}}$ is dense in $T_{\bar{X}}^{Cl}(\overline{\boldsymbol{x}})$, then $\overline{\boldsymbol{x}}$ is a Pareto-Clarke critical point for Problem (4.5). If, in addition, $\bar{\mathbf{F}}$ is strictly differentiable at $\overline{\boldsymbol{x}}$, then this point is a Pareto-Clarke-KKT critical point for Problem (4.5).*

Proof. The proof follows from Theorem 4.2.8, using similar arguments to the ones of Theorem 4.9 in [64]. □

The previous theorem states that DMS-FILTER-IR generates a Pareto-Clarke critical point for Problem (4.5). However, if $\overline{\boldsymbol{x}}$ is a local efficient point of Problem (4.5) and $h(\overline{\boldsymbol{x}}) = 0$, then $\overline{\boldsymbol{x}}$ is a local efficient point for Problem (4.1).

Thus, a question that arises is if the algorithm is indeed able to generate feasible points, when initialized from infeasible ones. The next theorem attempts to establish some conditions that provide an answer to this question.

Theorem 4.2.10. *Let Assumption 4.2.1 hold. Assume that h is a continuous function and consider $\{\boldsymbol{x}_k\}_{k\in K}$, an infeasible refining subsequence, such that for each $k \in K$, $\boldsymbol{x}_k$ is used at a successful inexact restoration step in DMS-FILTER-IR. Then, the algorithm generates a limit point $\bar{\boldsymbol{y}} \in \Upsilon$.*

Proof. For each $k \in K$, $\boldsymbol{x}_k$ was used at a successful inexact restoration step. Thus, there is $\boldsymbol{y}_k^* \in X$ such that y_k^* is nondominated and $0 \leq h(\boldsymbol{y}_k^*) \leq \xi(\alpha_k)h(\boldsymbol{x}_k) \leq h_{\texttt{max}}$.

Since $\{\boldsymbol{x}_k\}_{k\in K}$ is a refining subsequence, $\lim_{k\in K}\alpha_k = 0$. Considering the properties of $\xi(\cdot)$ and the boundedness of h, we can conclude $\lim_{k\in K} h(\boldsymbol{y}_k^*) = 0$. Assumption 4.2.1 allows to consider $K^{\bar{\boldsymbol{y}}} \subseteq K$ such that $\lim_{k\in K}\boldsymbol{y}_k^* = \bar{\boldsymbol{y}}$ and the continuity of h establishes $h(\bar{\boldsymbol{y}}) = 0$, meaning that $\bar{\boldsymbol{y}} \in \Upsilon$. □

The following result links sequences of points generated by DMS-FILTER-IR to Problem (4.1). For establishing it, we will assume that globalization is based on the use of integer lattices.

Corollary 4.2.11. *Consider $\{\boldsymbol{x}_k\}_{k\in K}$ a feasible refining subsequence, converging to the refined point $\overline{\boldsymbol{x}} \in \Upsilon$, generated by algorithm DMS-FILTER-IR, when using a globalization strategy based on integer lattices. Assume that $\bar{\mathbf{F}}$ is Lipschitz continuous near $\overline{\boldsymbol{x}}$. Let $\boldsymbol{d} \in H_{\Upsilon}^{Cl}(\overline{\boldsymbol{x}})$ be a refining direction for $\overline{\boldsymbol{x}}$, associated with the refining subsequence $\{\boldsymbol{x}_k\}_{k\in K}$. Then,*

$$\exists \ell \in \{1, \ldots, m\} : f_{\ell}^{\circ}(\overline{\boldsymbol{x}}; \boldsymbol{d}) \geq 0.$$

Proof. The proof follows directly from the proof of Theorem 4.2.8, considering $\boldsymbol{d} \in H_{\Upsilon}^{Cl}(\overline{\boldsymbol{x}})$ and noting that for k sufficiently large $h(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) = 0 = h(\boldsymbol{x}_k)$. Thus, inequality (4.6) needs to hold for $\ell \in \{1, \ldots, m\}$. □

In this situation, $\overline{\boldsymbol{x}}$ is a Pareto-Clarke critical point for Problem (4.1).

Corollary 4.2.12. *Consider $\{\boldsymbol{x}_k\}_{k\in K}$ a feasible refining subsequence, converging to the refined point $\overline{\boldsymbol{x}} \in \Upsilon$, generated by algorithm DMS-FILTER-IR, when using a globalization strategy based on integer lattices. Assume that $H_{\Upsilon}^{Cl}(\overline{\boldsymbol{x}}) \neq \emptyset$ and that $\bar{\mathbf{F}}$ is Lipschitz continuous near $\overline{\boldsymbol{x}}$. If the set of refining directions for $\overline{\boldsymbol{x}}$ is dense in $T_{\Upsilon}^{Cl}(\overline{\boldsymbol{x}})$, then $\overline{\boldsymbol{x}}$ is a Pareto-Clarke critical point for Problem (4.1). If, in addition, $\bar{\mathbf{F}}$ is strictly differentiable at $\overline{\boldsymbol{x}}$, then this point is a Pareto-Clarke-KKT critical point for Problem (4.1).*

Proof. The proof is a direct consequence of Corollary 4.2.11, using similar arguments to the ones of Theorem 4.9 in [64]. □

By using a filter-based approach, we have overcome the difficulty of applying directional direct search to multiobjective constrained optimization problems, when a feasible initialization is not available (regarding the relaxable constraints). The incorporation of a inexact restoration step potentiates feasibility. The next section will illustrate the numerical competitiveness of the proposed approach.

4.3 Implementation details

Differently from DMultiMADS-PB [29], DMS-FILTER-IR selects a single point at each iteration, to be explored at the search and, possibly, at the inexact restoration and/or pool steps. The decision on using a feasible or infeasible iterate point always attempts to promote feasibility, regarding the relaxable constraints (unrelaxable constraints are always satisfied by the points in the list, from where the iterate point will be selected).

The algorithm switches to an infeasible iterate point if the current feasible iterate point only generates infeasible points. At the next iteration, the iterate point will be selected from the nondominated points in the list that do not satisfy the relaxable constraints. Once

that an infeasible iterate point generates at least one feasible point, a feasible point will be selected as iterate point for the next iteration.

Suppose that we are at one iteration where the iterate point should be feasible. For selecting it from all the feasible points in the iterate list, we use the concept of the most isolated point. For each component of the objective function $\ell = 1, \ldots, m$, the feasible points in L_k are selected and ordered by increasing function value:

$$f_\ell(\boldsymbol{x}^1) \leq f_\ell(\boldsymbol{x}^2) \leq \ldots \leq f_\ell(\boldsymbol{x}^{m_F}),$$

where m_F denotes the total number of feasible points in L_k.

Then, for each component of the objective function f_ℓ, $\ell = 1, \ldots, m$, and for each feasible point $\boldsymbol{x}^j$ in L_k, with $j = 1, \ldots, m_F$, the following indicator is computed:

$$\delta_\ell\left(\boldsymbol{x}^j\right) = \begin{cases} f_\ell\left(\boldsymbol{x}^2\right) - f_\ell\left(\boldsymbol{x}^1\right) & \text{, if } j = 1 \\ f_\ell\left(\boldsymbol{x}^{m_F}\right) - f_\ell\left(\boldsymbol{x}^{m_F-1}\right) & \text{, if } j = m_F \\ \dfrac{f_\ell\left(\boldsymbol{x}^{j+1}\right) - f_\ell\left(\boldsymbol{x}^{j-1}\right)}{2} & \text{, otherwise.} \end{cases}$$

The most isolated point corresponds to the maximum value of γ_j, that is,

$$\boldsymbol{x}_k \in \underset{j=1,\ldots,m_F}{\arg\max}\, \gamma_j,$$

where

$$\gamma_j = \frac{1}{m} \sum_{\ell=1}^{m} \delta_\ell(\boldsymbol{x}^j). \tag{4.7}$$

When the iterate point should be infeasible, two different criteria are used for its selection, depending on having at least one feasible point in the list or not. In the latter situation, the point selected corresponds to the most promising infeasible point is the list, in terms of restoring the feasibility. Thus, the point with the smallest value for the aggregated constraint violation function h will be chosen.

Now, suppose that there is at least one feasible point in the list. The fact that the iterate point should be infeasible means that at the last iteration a feasible iterate point only generated infeasible ones. Thus, we want to try to restore feasibility close to the region that was being explored. A closed ball centered on the feasible iterate point, of radius equal to $\eta\, \alpha_k \max_{\boldsymbol{d} \in D_{k-1}} \|\boldsymbol{d}\|$, with $\eta \geq \frac{1}{\beta_1}$, is considered and the infeasible point in the list, belonging to this ball, with the lowest value for the aggregated constraint violation function h will be selected as iterate point.

DMS-FILTER-IR was implemented in `Matlab`, keeping the default settings of DMS. Thus, the stepsize was initialized as 1, kept constant at successful iterations ($\gamma = 1$), and halved at unsuccessful ones ($\beta_1 = \beta_2 = 0.5$). No search step was implemented.

The aggregated violation function was defined as:

$$h(\boldsymbol{x}) = \| \mathbf{g}(\boldsymbol{x})_+ \|_2^2 = \sum_{i=1}^{p} \max\{0, g_i(\boldsymbol{x})\}^2.$$

Regarding the maximum violation allowed, $h_{\max}$, it depends on the initialization. If there are any infeasible points in the list, $h_{\max}$ will be set equal to the largest of the existing values of h. Otherwise, it will be set equal to the maximum between 10 and half of the number of the relaxable constraints.

Many options can be considered for defining a function complying with the requirements of $\xi(\cdot)$, to be used in the inexact restoration step. In the numerical implementation, we used the function $\xi(\alpha) = (\frac{\alpha}{2})^2$. Note that this function is continuous, $\xi(\alpha) \to 0$ when $\alpha \downarrow 0$ and, considering the initialization and the strategy to update the stepsize, $0 < \xi(\alpha) < 1$. The inexact restoration problems were solved with the `Matlab` function `fmincon.m`. Through the numerical section, for any solver, feasibility is assumed to be achieved when there is an aggregated violation of the relaxable constraints less than 10^{-5}.

If the poll step is performed, a complete polling approach is adopted, evaluating all the points corresponding to directions in the positive spanning set.

4.4 Numerical experiments

This section is devoted to the numerical experiments performed with DMS-FILTER-IR, illustrating its numerical behavior by comparison with the original DMS algorithm, when solving constrained problems, or with other state-of-art solvers. A first subsection will detail the problem collection. The last two subsections illustrate the numerical behavior of the proposed algorithm. All tests were performed in a laptop with a 11th Gen Intel® Core(TM) i7-1165G7 processor, at 2.80GHz, with 16GB of RAM memory, using Windows 11 with 64 bits.

4.4.1 Problem collection

Liuzzi et al. [134] defined a collection of constrained problems by coupling a subset of the bound constrained problems provided in Custódio et al. [64] with six families of constraints proposed in [119]. All the bound problems in [64] with $n \geq 3$ variables were selected, resulting in 51 bound constrained problems. A set of 306 constrained problems was generated by adding to each problem the following six families of nonlinear constraints (the suggested initialization is denoted by $\boldsymbol{x}^0$):

$$
\begin{aligned}
g_i^1(\boldsymbol{x}) &= (3 - 2\boldsymbol{x}_{i+1})\boldsymbol{x}_{i+1} - \boldsymbol{x}_i - 2\boldsymbol{x}_{i+2} + 1, \quad i = 1, \ldots, p, \quad p = n - 2 \\
&\quad \boldsymbol{x}_j^0 = 1, \quad j = 1, \ldots, n \\
g_i^2(\boldsymbol{x}) &= (3 - 2\boldsymbol{x}_{i+1})\boldsymbol{x}_{i+1} - \boldsymbol{x}_i - 2\boldsymbol{x}_{i+2} + 2.5, \quad i = 1, \ldots, p, \quad p = n - 2 \\
&\quad \boldsymbol{x}_j^0 = 2, \quad j = 1, \ldots, n \\
g_i^3(\boldsymbol{x}) &= \boldsymbol{x}_i^2 + \boldsymbol{x}_{i+1}^2 + \boldsymbol{x}_i\boldsymbol{x}_{i+1} - 2\boldsymbol{x}_i - 2\boldsymbol{x}_{i+1} + 1, \quad i = 1, \ldots, p, \quad p = n - 1 \\
&\quad \boldsymbol{x}_j^0 = 0.5 \quad j = 1, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
g_i^4(\boldsymbol{x}) &= \boldsymbol{x}_i^2 + \boldsymbol{x}_{i+1}^2 + \boldsymbol{x}_i\boldsymbol{x}_{i+1} - 1, \quad i = 1, \ldots, p, \quad p = n - 1 \\
&\quad \boldsymbol{x}_j^0 = 0, \quad j = 1, \ldots, n \\
g_i^5(\boldsymbol{x}) &= (3 - 0.5\boldsymbol{x}_{i+1})\boldsymbol{x}_{i+1} - \boldsymbol{x}_i - 2\boldsymbol{x}_{i+2} + 1, \quad i = 1, \ldots, p, \quad p = n - 2, \\
&\quad \boldsymbol{x}_j^0 = 2, \quad j = 1, \ldots, n \\
g_i^6(\boldsymbol{x}) &= \sum_{j=1}^{n-2}((3 - 0.50\boldsymbol{x}_{j+1})\boldsymbol{x}_{j+1} - \boldsymbol{x}_j - 2\boldsymbol{x}_{j+2} + 1), \quad i = p, \quad p = 1 \\
&\quad \boldsymbol{x}_j^0 = 2, \quad j = 1, \ldots, n
\end{aligned}
$$

In theory, DMS is developed for multiobjective derivative-free optimization, regardless of the number of components in the objective function. However, the numerical results reported in [64] respect to biojective and triobjective problems (there is an exception of one problem with four components in the objective function, from the set of 100 bound constrained problems considered). In fact, it is common knowledge in the multiobjective optimization community that addressing problems with more than three components in the objective function requires special techniques, falling on the specific domain of Many-objective Optimization [81]. The focus of this work is not Many-objective Optimization problems. Considering that the aggregated penalization function will be an additional component of the objective function, we restricted the set of 306 constrained problems to biobjective ones, in a total of 156. The problems and their dimensions are given in Table 4.1.

Problem	**n**	**Problem**	**n**	**Problem**	**n**
CL1	4	L2ZDT6	10	QV1	10
DPAM1	10	L3ZDT1	30	ZDT1	30
FES1	10	L3ZDT2	30	ZDT2	30
Kursawe	3	L3ZDT3	30	ZDT3	30
L1ZDT4	10	L3ZDT4	30	ZDT4	10
L2ZDT1	30	L3ZDT6	10	ZDT6	10
L2ZDT2	30	MOP2	4	SK2	4
L2ZDT3	30	MOP4	3	TKLY1	4
L2ZDT4	30	OKA2	3		

Table 4.1: The biobjective test set used in the numerical experiments (n denotes the number of variables).

DMS requires a feasible initialization. However, not all the points $\boldsymbol{x}^0$ provided by Karmitsa [119] satisfy the bounds constraints. Thus, from the 156 biobjective problems considered, we retained only the 93 for which a feasible initialization was available. The final test set comprised 93 nonlinearly constrained biobjective problems, with $n \in [3, 30]$ and $p \in [1, 29]$. We assumed the nonlinear constraints as being relaxable, corresponding the unrelaxable constraints to bounds.

4.4.2 Positive spanning sets

The convergence of both DMS and DMS-FILTER-IR is established under the assumption of asymptotic density of the sets of directions used by the algorithm, during the optimization process. However, when the budget of function evaluations is limited, which is often the case when the function is expensive to evaluate, this density is never accomplished. Coordinate search has the perfect geometry for bound constrained problems. In fact, this was the type of directions used to obtain the numerical results reported in [64], illustrating the numerical performance of DMS.

Considering that in DMS-FILTER-IR the nonlinear constraints are addressed by the filter approach, using the aggregated violation function and reducing the problem to a bound constrained problem, it would be interesting to compare DMS and DMS-FILTER-IR when using coordinate search as set of directions or when resorting to an asymptotic dense set of directions, built using the technique proposed in [1], based on Halton sequences.

DMS addresses constraints with an extreme barrier approach, only evaluating feasible points and requiring a feasible initialization. Thus, the feasible point x^0 given in [119] was used for initialization. DMS-FILTER-IR allows infeasible points, with respect to the relaxable constraints. Therefore, DMS-FILTER-IR was initialized with n-points equally spaced in the line segment, joining the variable upper and lower bounds.

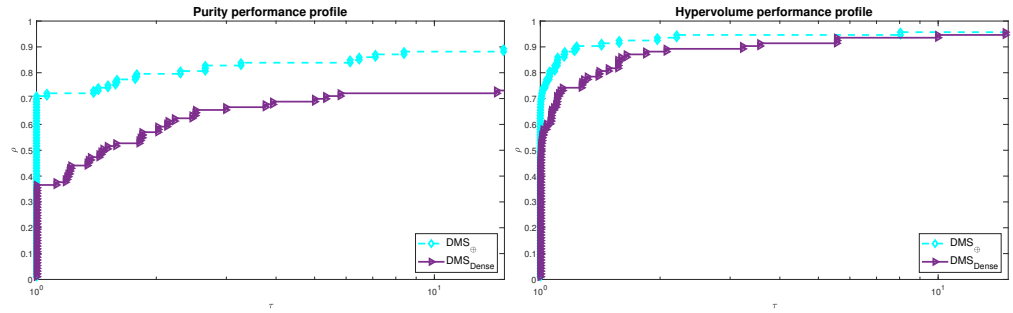
Figures 4.2 and 4.3 report the results obtained for DMS and DMS-FILTER-IR, respectively. A maximum budget of 5000 function evaluations was allowed, jointly with a minimum stepsize of 10^{-3}, for all the points in the list.

With exception to the spread metric Γ, where the results are very close, for any of the metrics considered, it is clear the advantage of the use of coordinate directions as positive spanning set, both for DMS and DMS-FILTER-IR. As already mentioned, this could be the result of the nice geometry associated to these directions and bound constrained problems. Thus, in the next section, DMS and DMS-FILTER-IR will consider positive spanning sets based on coordinate search.

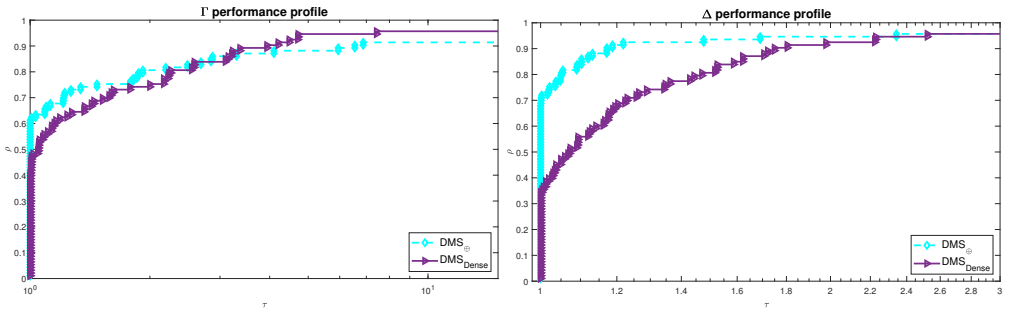
4.4.3 Comparison with other solvers

In addition to DMS [64], DFMO [134], and DMultiMADS-PB [29] were also tested as benchmark solvers to evaluate the performance of DMS-FILTER-IR. The DMS solver is implemented in `Matlab` and is freely available at `http://www.mat.uc.pt/dms`. DFMO is coded in `Fortran90` and is available at `https://github.com/DerivativeFreeLibrary`. Finally, coded in `Julia`, DMultiMADS-PB can be obtained from `https://github.com/bbopt/DMultiMadsPB`.

The DFMO and the DMultiMADS-PB algorithms address nonlinearly constrained multiobjective optimization problems by penalizing the nonlinear constraints with an exact merit function or by using a progressive barrier approach, respectively. As already mentioned, DMS addresses constraints with an extreme barrier function.

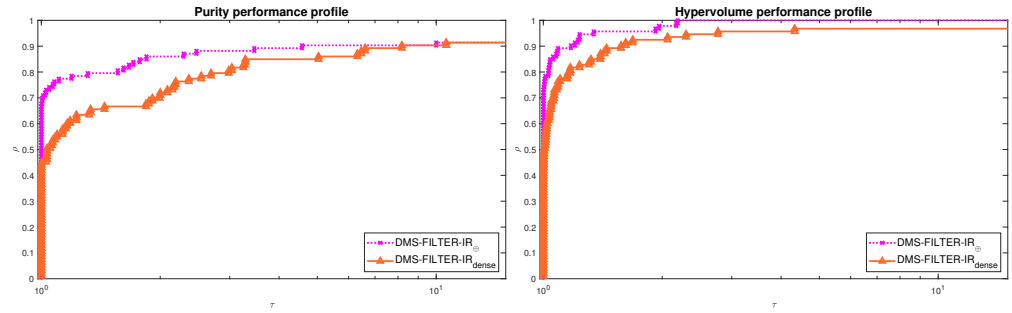


(a) Purity (b) Hypervolume

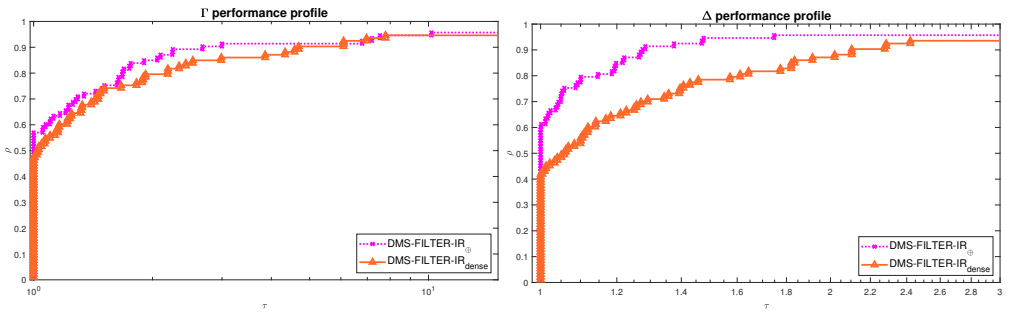


(c) Spread Gamma (d) Spread Delta

Figure 4.2: Performance profiles for DMS, considering different types of positive spanning sets and a maximum budget of 5000 function evaluations.



(a) Purity (b) Hypervolume



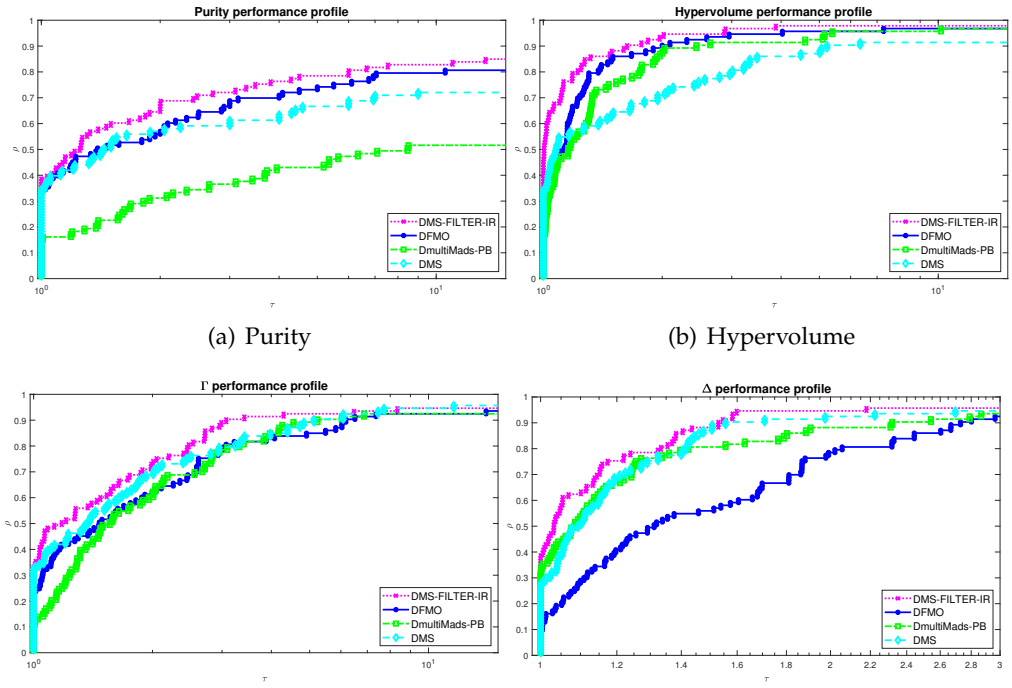
(c) Spread Gamma (d) Spread Delta

Figure 4.3: Performance profiles for DMS-FILTER-IR, considering different types of positive spanning sets and a maximum budget of 5000 function evaluations.

All solvers were run with the defaults, selecting the best version identified in [134] for DFMO and using the best version reported in [29] for DMultiMADS-PB. Results were obtained for maximum budgets of 500 and 5000 function evaluations. Considering the expensive nature of function evaluation, small budgets are particular relevant for assessing the performance of the solvers.

As in the previous subsection, DMS was initialized with the feasible point x^0 provided in [119]. DMS-FILTER-IR, DMultiMADS-PB, and DFMO can be initialized with infeasible points. Therefore, DMS-FILTER-IR and DMultiMADS-PB were initialized with n-points equally spaced in the line segment, joining the variable upper and lower bounds, which is the default initialization of DMultiMADS-PB. DFMO was initialized with the centroid of the box defined by the bound constraints. After, the algorithm also generates n-points equally spaced in the line segment joining the bounds.

Figure 4.4 depicts performance profiles for the different metrics considered, when a maximum budget of 500 function evaluations is allowed.



(a) Purity (b) Hypervolume

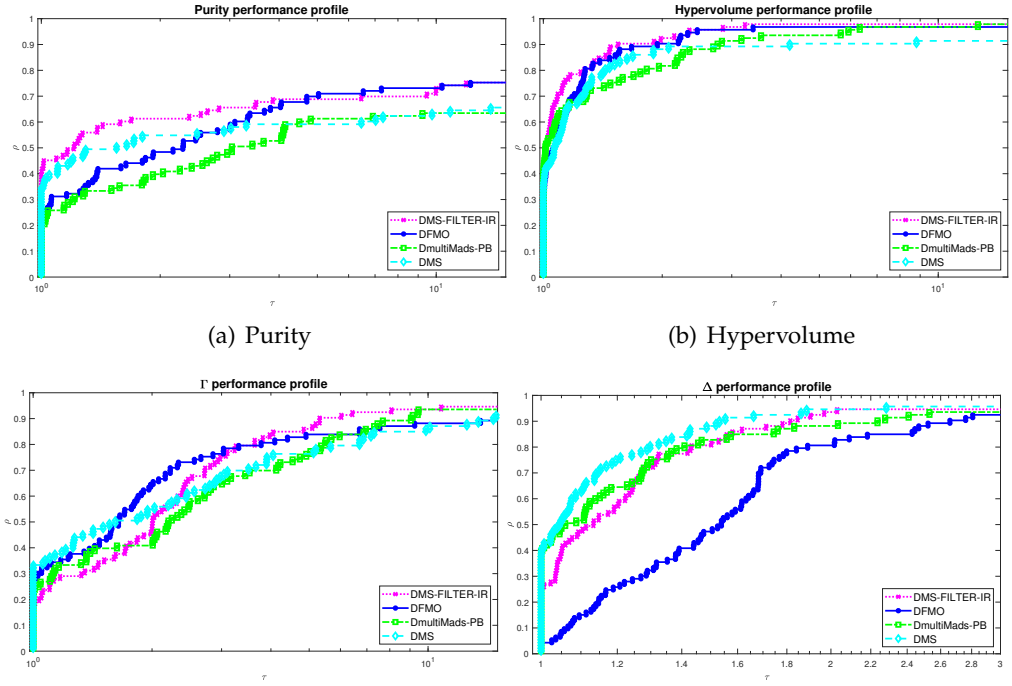
(c) Spread Gamma (d) Spread Delta

Figure 4.4: Comparing DMS-FILTER-IR with DFMO, DMultiMADS-PB, and DMS based on performance profiles for a maximum of 500 function evaluations.

In general, DMS-FILTER-IR presents a good performance for any of the four metrics considered. Noteworthy, it is the most efficient solver for hypervolume and presents some advantage regarding robustness for the purity metric.

When the maximum budget allowed increases to 5000 function evaluations (see Figure 4.5), DMS-FILTER-IR remains as the most competitive solver in what respects purity and hypervolume. For the spread metric Δ, DMS presents a better performance. DFMO

provides some good results in terms of the largest gap in the Pareto front, represented by the metric Γ.



(a) Purity

(b) Hypervolume

(c) Spread Gamma

(d) Spread Delta

Figure 4.5: Comparing DMS-FILTER-IR with DFMO, DMultiMADS-PB, and DMS based on performance profiles for a maximum of 5000 function evaluations.

Individual comparisons between DMS-FILTER-IR and each of the three remaining solvers are provided in the Appendix A, specifically in Figures A.1 and A.2. These figures present results for budgets of 500 and 5000 function evaluations, respectively, offering further clarification on the previous analysis and supporting the conclusions drawn.

5

A Direct Multisearch Approach for Many-Objective DFO

The strategy proposed in Chapter 4 to address nonlinear constraints in MOO can be applied to any problem, regardless of the number of components in the objective function. However, in Section 4.4, results were only reported for biobjective problems.

In fact, addressing constraints with the filter approach for problems with more than two components in the objective function results in a so-called many-objective optimization problem [81]. Even if the theory supports it, DMS [64] was never numerically tested in a many-objective optimization framework.

As the number of components of the objective function increases, it becomes much simpler to find new nondominated points, resulting in the proliferation of nondominated solutions. This phenomenon underscores the complexity inherent to the many-objective optimization class, currently typically addressed with heuristics (see [131] for a recent survey on the subject).

In a recent work [171], Roberts and Royer presented a study on a direct search algorithm based on random subspaces of arbitrary dimension. The complexity guarantees of this approach were established based on the probabilistic properties of both the subspaces and the directions within them. Unlike previous analyses provided in [107], the study relaxes the requirement for direction norms to be deterministically bounded, thus allowing for a broader range of optimal polling strategies, characterized by these properties. By leveraging insights from random subspace embeddings and sketching matrices, complexity bounds were improved for randomized instances of the proposed algorithmic framework.

In this chapter, we present a new variant of Direct Multisearch, named DMS-Reduction, which relies on search directions chosen within subspaces of the variables space, as proposed in [171], or is based on correlation matrices that measure the correlation between variables and objectives. By selecting appropriated subsets of components of the objective function to be optimized at each iteration, the number of variables to be optimized can also be decreased, as not all variables may be related to the selected objective function components. Thus, DMS-Reduction is designed to address large-scale MOO problems,

concerning the number of components of the objective function and the number of problem variables.

In Section 5.1, we describe the proposed algorithmic structure. Section 5.2 details the numerical implementation used for the extensive numerical experiments reported in Section 5.3.

5.1 Algorithmic structure of DMS-Reduction

We address the unconstrained many-objective optimization problem defined by:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))^\top \tag{MOO}$$

where $\boldsymbol{F} : \mathbb{R}^n \to \{\mathbb{R} \cup \{+\infty\}\}^m$, with $m \geq 4$.

Since DMS-Reduction is developed within the DMS framework, a list of nondominated points and corresponding stepsizes is maintained throughout the iterations. Each iteration aims at improving this list by adding new nondominated points, while removing dominated ones.

At each iteration, a point and its corresponding stepsize are selected from the list. Subsequently, an optional search step and a poll step are performed. The difference with the classical DMS algorithm relies on the poll step. An adequate matrix $\boldsymbol{P} \in \mathbb{R}^{r\times n}$ is used to move the optimization to a reduced subspace. Poll directions are considered in the reduced subspace, trial poll points are generated and tested for nondominance, this time in the original spaces of n variables and m components of the objective function. The remaining algorithmic procedures mimetize the original DMS.

Algorithm 8 formalizes the DMS-Reduction framework.

Definition 5.1.1. *Let $\boldsymbol{y} \in \mathbb{R}^n$ and L be a list of nondominated points. We say that $\boldsymbol{y}$ is dominated if:*

$$\exists \boldsymbol{x} \in L : F(\boldsymbol{x}) - \rho(\alpha) \leq F(\boldsymbol{y}),$$

where $\rho(\cdot)$ denotes a forcing function and α the stepsize associated to the current iteration.

Figure 5.1 illustrates the situation for a biobjective problem ($m = 2$), when the current list of nondominated points is formed by two points, in red, $L = \{\boldsymbol{x}, \boldsymbol{z}\}$, whose images by function $\boldsymbol{F}$ correspond to the red dots. The set $D(L) \subset \mathbb{R}^m$ represents the image of the set of points dominated (in the classical Pareto sense) by the points in L and $D(L; \rho(\alpha))$, in blue or red, denotes the set of points whose distance in the ℓ_∞ norm to $D(L)$ is no larger than $\rho(\alpha) > 0$. Note that $D(L) \subset D(L; \rho(\alpha))$. A new point $\boldsymbol{y} \in \mathbb{R}^n$ will be accepted if $F(\boldsymbol{y}) \notin D(L; \rho(\alpha))$, ensuring an increase in the hypervolume associated to the list of points of at least $(\rho(\alpha))^m$ (see in [63, Lem. 3.1]). Therefore, new nondominated points will be accepted if they are in the green area.

Algorithm 8 DMS-Reduction for Many-Objective Optimization

Initialization:

Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$ with $f_\ell(\boldsymbol{x}_0) < +\infty \; \forall \ell \in \{1, \ldots, m\}$, $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, $\gamma \geq 1$, and $r < n$. Initialize the list of nondominated points and corresponding stepsize parameters $L_0 = \{(\boldsymbol{x}_0; \alpha_0)\}$.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according some criteria and select the first item $(\boldsymbol{x}; \alpha) \in L_k$ as the current iterate and stepsize parameter (thus setting $(\boldsymbol{x}_k; \alpha_k) = (\boldsymbol{x}; \alpha)$).

2. **Search step:** Compute a finite set of points $\{\boldsymbol{z}_s\}_{s \in S}$ and evaluate $\boldsymbol{F}$ in each element of $\{\boldsymbol{z}_s\}_{s \in S}$. Set $L_{\texttt{add}} = \{(\boldsymbol{z}_s; \alpha_k), s \in S\}$. Call $L_{\texttt{filtered}} = \texttt{filter}(L_{\texttt{add}}, L_k)$ to remove all dominated points from $L_k \cup L_{\texttt{add}}$. If $L_{\texttt{filtered}} \neq L_k$, declare the iteration as successful, set $L_{k+1} = L_{\texttt{filtered}}$, and skip the poll step.

3. **Poll step:** Compute an adequate matrix $\boldsymbol{P}_k \in \mathbb{R}^{r \times n}$ and a set of poll directions $\mathcal{D}_k$ in $\mathbb{R}^r$. Evaluate $\boldsymbol{F}$ at the set of poll points $P_k = \{\boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^\top \boldsymbol{d} \mid \boldsymbol{d} \in \mathcal{D}_k\}$. Set $L_{\texttt{add}} = \{(\boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^\top \boldsymbol{d}; \alpha_k), \boldsymbol{d} \in \mathcal{D}_k\}$. Call $L_{\texttt{filtered}} = \texttt{filter}(L_{\texttt{add}}, L_k)$ to remove all dominated points from $L_k \cup L_{\texttt{add}}$. If $L_{\texttt{filtered}} \neq L_k$, declare the iteration as successful and $L_{k+1} = L_{\texttt{filtered}}$. Otherwise, declare the iteration as unsuccessful and set $L_{k+1} = L_k$.

4. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma \alpha_k]$. Replace all the new points $(\boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^\top \boldsymbol{d}; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k^\top \boldsymbol{d}; \alpha_{k,\texttt{new}})$, when success is coming from the poll step, or $(\boldsymbol{z}_s; \alpha_k)$ in L_{k+1} by $(\boldsymbol{z}_s; \alpha_{k,\texttt{new}})$, when success is coming from the search step. Replace also $(\boldsymbol{x}_k; \alpha_k)$, if in L_{k+1}, by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$. Otherwise, decrease the stepsize parameter, by choosing $\alpha_{k,\texttt{new}} \in [\beta_1 \alpha_k, \beta_2 \alpha_k]$, and replace the poll pair $(\boldsymbol{x}_k; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$.

EndFor

5.2 Implementation details

Before establishing the theoretical properties of Algorithm 8, we decided to evaluate its numerical performance, under different options for matrices $\boldsymbol{P}_k$. For that, a bound constrained MOO setting was considered, taking advantage of problems that are part of the collection reported in [64], that allow a variable dimension, both at the objectives and variables spaces. This section details the practical implementation of Algorithm 8.

The default initialization of DMS consists of n-points equally spaced along the line segment that joins the variable upper and lower bounds. However, in the initialization phase of Algorithm 8, the initial list of nondominated points, denoted as L_0, is computed using Algorithm 9, including the extreme points of the problem, corresponding to the individual minimization of each function component, and, optionally, the middle points, detailed after.

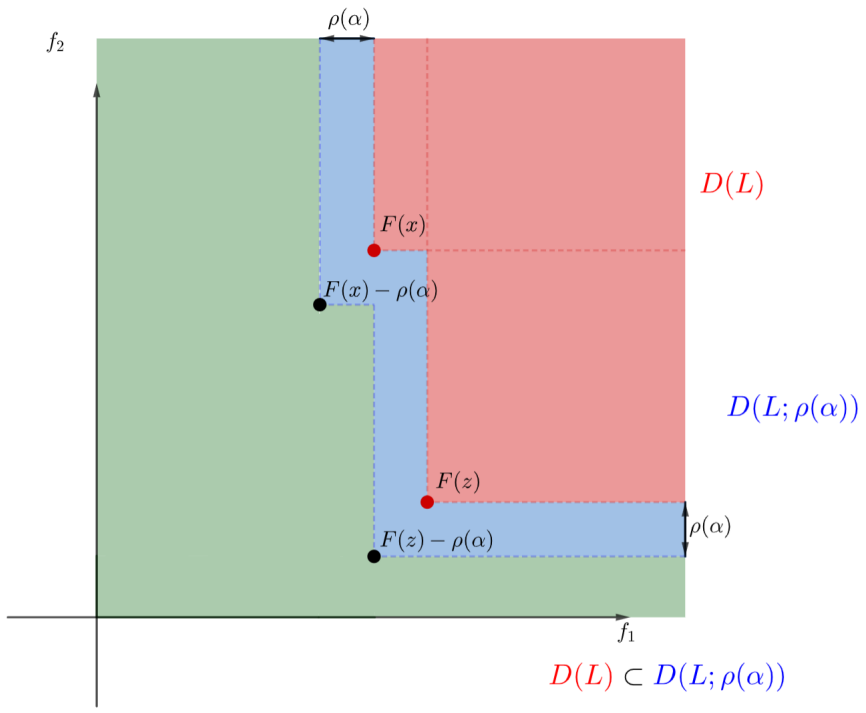


Figure 5.1: Globalization strategy based on a sufficient decrease condition, with hypervolume increase marking a successful iteration.

Algorithm 9 Extreme Points & Middle Points

Extreme Points: For each $\ell \in \{1, \ldots, m\}$ compute the extreme point given by

$$\overline{x}^{\ell} = \arg\min_{x \in \mathbb{R}^n} f_{\ell}(x).$$

Middle Points (Optional): For each integer $\xi \in \{2, \ldots, m\}$, consider the set of combinations of ξ components of objective function (which has $\binom{m}{\xi}$ elements). Let J_{ξ}^{ℓ} be one of its elements. Compute the middle point as

$$\hat{x}^{\ell} = \frac{1}{\xi} \sum_{j \in J_{\xi}^{\ell}} \overline{x}^{j}, \quad \ell = 1, \ldots, \binom{m}{\xi}.$$

Define the initial list of points, L_0, as

$$L_0 := \{(\overline{x}^{\ell}; \alpha_0), \ell = 1, \ldots, m\} \cup \left\{ \bigcup_{\xi=2}^{m} \bigcup_{\ell=1}^{\binom{m}{\xi}} \{(\hat{x}^{\ell}; \alpha_0)\} \right\}.$$

The extreme points are obtained using the SID-PSM algorithm [66, 65]. For each run of SID-PSM, the stepsize is initialized as $\max\{1, \|x_{\text{initial}}\|_{\infty}\}$, with x_{initial} being the centroid of the box defined by the bound constraints. The lowest value allowed for the

stepsize parameter is 10^{-3}, equal to the value of DMS, and a maximum of 500 function evaluations is allowed. When accounting these function evaluations in the total budget used by DMS-Reduction, we calculate the average of the budget spent in the extreme points computation, rounding up, since at each iteration of SID-PSM only f_ℓ, for $\ell \in \{1, \ldots, m\}$, rather than $\boldsymbol{F}$, is computed. The final evaluation of the vector function $\boldsymbol{F}$ is performed once for each extreme point obtained, and these m evaluations are also accounted in the DMS-Reduction budget.

Considering the extension anticipated for the Pareto front, the use of an initialization that considers extreme points could potentially help the algorithm to expand the approximation to the Pareto front. The middle points correspond to additional points, trying to obtain a good initial cover of the domain.

As example, in Figure 5.2, we present a straightforward tri-objective problem. The Pareto front takes the form of a simplex, defined as $\{\boldsymbol{y} \in \mathbb{R}^3_+ \mid \sum_{i=1}^3 y_i = 1\}$. The black points represent the extreme points, corresponding to each one of the three objectives. The blue points represent the middle points, taking into consideration the pairs (f_1, f_2), (f_1, f_3), and (f_2, f_3). Lastly, the red point represents the middle point when considering all three objectives simultaneously.

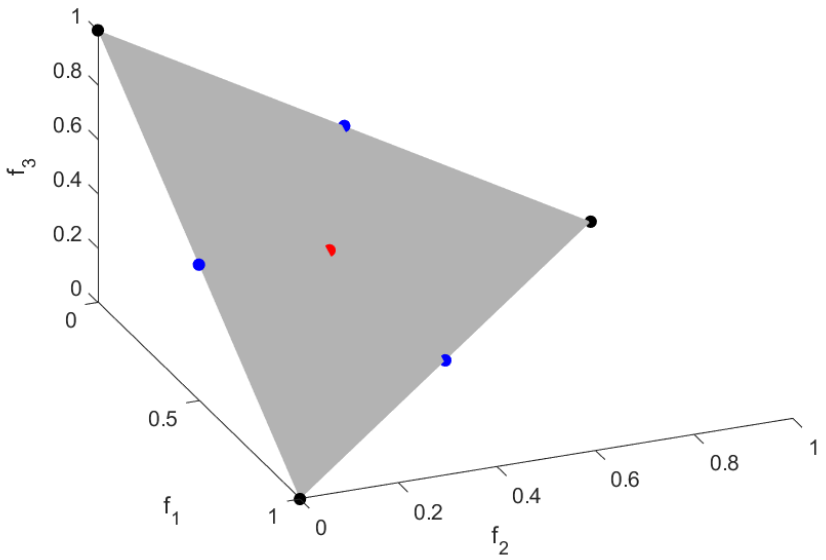


Figure 5.2: Simple example of Extreme & Middle Points computation in a tri-objective problem.

New strategies were also considered in DMS-Reduction for the selection of an iterate center, which occurs in Step 1 of Algorithm 8. Since the performance of the original DMS tends to deteriorate with the increase of the number of components of objective function, our criterion for selecting an incumbent point focuses only on a subset of objectives. By working with this subset of objectives in each iteration, and selecting as iterate point, the

one corresponding to the largest value of the Γ metric for the objectives selected, we aim at exploring different parts of the nondominated set, while attempting to fill gaps within the subset. However, this process is only applicable to the selection of the iterate center. Anytime that dominance is checked, either at the search or poll steps, all the m objectives are taken into account.

For the selection of the subset of objectives to be considered, three criteria were used: the most conflicting, random, or cyclic approaches. The most conflicting objectives are determined using a correlation matrix, which may be computed based on either the Spearman or the Kendall coefficients [97, Ch. 11]. Both types of correlations are non-parametric measures, not requiring particular assumptions on the distribution of the data, as a linear relationship, and range from -1 to $+1$.

In order to compute the Kendall's correlation, denoted by τ, between two variables $X, Y \in \mathbb{R}^n$, consider $(X_1, Y_1), \ldots, (X_n, Y_n)$, n i.i.d. samples of (X, Y). Then:

$$\tau = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \operatorname{sgn}\left[(X_i - X_j)(Y_i - Y_j)\right],$$

where $\operatorname{sgn}(\cdot)$ represents the sign function.

In this case, a value of $\tau = -1$ indicates discordance, meaning that when one variable increases, the other tends to decrease. In contrast, a value $\tau = 1$ indicates concordance, meaning that when one variable increases, the other tends also to increase. A value of $\tau = 0$ indicates no association between the variables.

The Spearman correlation, denoted by ρ, is equivalent to the commonly used Pearson's linear correlation, but applied to the ranks of samples. If all ranks associated to one variable are distinct, it simplifies to

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)},$$

where d_i is the difference between the ranks for each pair of observations.

The Spearman correlation coefficient assesses the monotonic relationship between two variables. Thus a value of $\rho = 1$ indicates a perfect positive correlation, meaning that as one variable increases, the other variable also increases. A value of $\rho = -1$ signifies a perfect negative correlation, where one variable increases as the other decreases. A correlation value of $\rho = 0$ indicates no correlation, implying that the variables are independent of each other.

For deciding the number of components of the objective function to be selected at each iteration, an easy choice is always consider biobjective problems, given the good performance of DMS on this class. More sophisticated approaches, based on the use of Principal Component Analysis (PCA), to dynamically select a variable number of objectives at each iteration, were considered, but with worse computational results. Thus, under the correlation approach, at each iteration, the two objectives with the most negative correlation are selected.

However, when there are specific pairs of objectives that demonstrate high levels of conflict, after several iterations, we may find ourselves repeatedly selecting the same pair of objectives. Employing random selection or cycling ensures that we avoid consistently selecting the same objectives, thus granting equal importance to each objective.

Note that reducing the number of components of the objective function to be optimized at each iteration has the additional benefit of possibly conducting to a reduction in the number of variables to be optimized at each iteration, since there could be the case that not all variables are related to the objective function components selected. Therefore, dimensionality reduction is also done in the poll step, focusing on reducing the number of variables.

This variable reduction can be done either through sketching approaches [43] or again using correlation, this time computed between the original n variables and the two objectives selected. A variable is kept at a given iteration if it has at least 10% of absolute value correlation with one of the selected objectives. Note that, in this case, we are considering the absolute values of the correlation, because the type of monotonicity is not relevant for the choice.

For sketching, following the approach used in [171], we can consider three possibilities for matrix $\boldsymbol{P}_k$:

- $\boldsymbol{P}_k$ has entries which are i.i.d. $\mathcal{N}(0, 1/r)$ with $r \in \mathbb{N}$ and $r < n$.
- $\boldsymbol{P}_k$ is a s-hashing matrix, i.e., every column of $\boldsymbol{P}_k$ has exactly $s \in \mathbb{N}$ nonzero entries at randomly selected locations, each taking value $\pm 1/\sqrt{s}$ with independent probability $1/2$.
- $\boldsymbol{P}_k = \sqrt{n/r}\mathbb{I}_{r\times n}\boldsymbol{Q}^\top$ where $\mathbb{I}_{r\times n}$ denotes the first r rows of the $n \times n$ identity matrix $\mathbb{I}_n$, and $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ is the orthogonal factor in the QR decomposition $\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R} \in \mathbb{R}^{n\times n}$ of a matrix $\boldsymbol{Z}$ with i.i.d. standard normal entries such that the diagonal entries of $\boldsymbol{R}$ are positive.

Algorithm 10 summarizes the approaches that can be used to compute the matrix $\boldsymbol{P}_k \in \mathbb{R}^{r\times n}$ and the set $\mathcal{D}_k \subset \mathbb{R}^r$.

Algorithm 10 Directions in the Reduced Space

If Correlation Approach: Using correlation, select the subset of $r < n$ variables to keep. Consider $\mathcal{D}_k = [\mathbb{I}_r \ -\mathbb{I}_r]$, where $\mathbb{I}_r$ denotes the $(r \times r)$ identity matrix. Compute the matrix $\boldsymbol{P}_k \in \mathbb{R}^{r\times n}$ by adding to $\mathbb{I}_r$ $n - r$ columns of zeros, at columns corresponding to the indexes of the variables removed.

If Sketching Approach: Generate a sketching matrix $\boldsymbol{P}_k \in \mathbb{R}^{r\times n}$, where $r < n$, utilizing one of the following methods: Gaussian, Orthogonal, or Hashing. Define the set of directions in the reduced space $\mathcal{D}_k = [\mathbb{I}_r \ -\mathbb{I}_r]$, where $\mathbb{I}_r$ denotes the $(r \times r)$ identity matrix.

5.3 Numerical experiments

In this section, our goal is to assess the numerical performance of DMS-Reduction. Comparisons were made not only between different variants of DMS-Reduction, but also with several established genetic/evolutionary algorithms designed for many-objective optimization, as outlined in Subsection 5.3.1. Furthermore, we evaluated the ability of DMS-Reduction to approximate Pareto fronts by applying it to a practical, real-world multiobjective derivative-free optimization problem, detailed in Subsection 5.3.2. All tests were conducted on a laptop with a 11th Gen Intel® Core(TM) i7-1165G7 processor, at 2.80GHz, with 16GB of RAM memory, using Windows 11 with 64 bits.

5.3.1 Many-objective problem collection

In order to access the performance of the algorithm, we selected two test sets commonly used in many-objective optimization, namely the DTLZ [73] and the WFG [116]. The DTLZ and WFG are scalable multiobjective test problems, that offer flexibility by accommodating various characteristics, such as bias, multi-modality, and non-separability. These problems support a diverse range of Pareto optimal geometries, encompassing convex, concave, mixed convex/concave, linear, degenerated, and disconnected geometries, providing a versatile platform for testing and evaluation.

In Table 5.1, we provide additional information about each problem. For DTLZ1-DTLZ6 and WFG1-WFG9, we considered a number of components of the objective function in the range $m = 4, \ldots, 10$. The corresponding number of decision variables and the type of shape of the true Pareto front are indicated in Table 5.1. Therefore, we have a final test problem collection with 105 test problems.

Test Problem	**Dimension –** n	**Shape of true Pareto Front**
DTLZ1	$m + 4$	Linear
DTLZ2-4	$m + 9$	Concave
DTLZ5	$m + 9$	Mixed Convex/concave
DTLZ6	$m + 19$	Mixed Convex/concave
WFG1	$2m + 18$	Convex
WFG2	$2m + 18$	Convex Disconnected
WFG3	$2m + 18$	Linear Degenerated
WFG4-9	$2m + 18$	Concave

Table 5.1: Set considered for many-objective optimization.

The first numerical results respect to the comparison between two variants of DMS-Reduction (one deterministic and the other stochastic), alongside with the original DMS:

- DMS-Reduction - Deterministic: Algorithm 8 with Algorithm 9 for initialization, and the correlation approach of Algorithm 10.
- DMS-Reduction - Stochastic: Algorithm 8 with Algorithm 9 for initialization, and the sketching approach of Algorithm 10.

Regarding DMS-Reduction-Deterministic and DMS-Reduction-Stochastic, several variants were tested. For the stochastic approach, the best version considers a random selection of the objectives and the utilization of a 1−hashing matrix to construct the reduced set of directions for the poll step. The deterministic version corresponds to the best version among all deterministic versions tested, which uses the Spearman correlation to select the objectives and also to select the subsets of variables.

For all the different versions, we set $\alpha_0 = 1$, $\beta_1 = \beta_2 = 0.5$, and $\gamma = 1$. Simple decrease was employed to accept new nondominated points, i.e., we have chosen $\overline{\rho}(\cdot)$ as the constant, zero vector of dimension m. As stopping criteria, we used a maximum stepsize in the list of 10^{-3} or a maximum of 5000 function evaluations. In addition, for stochastic solvers, the average of 10 runs was considered.

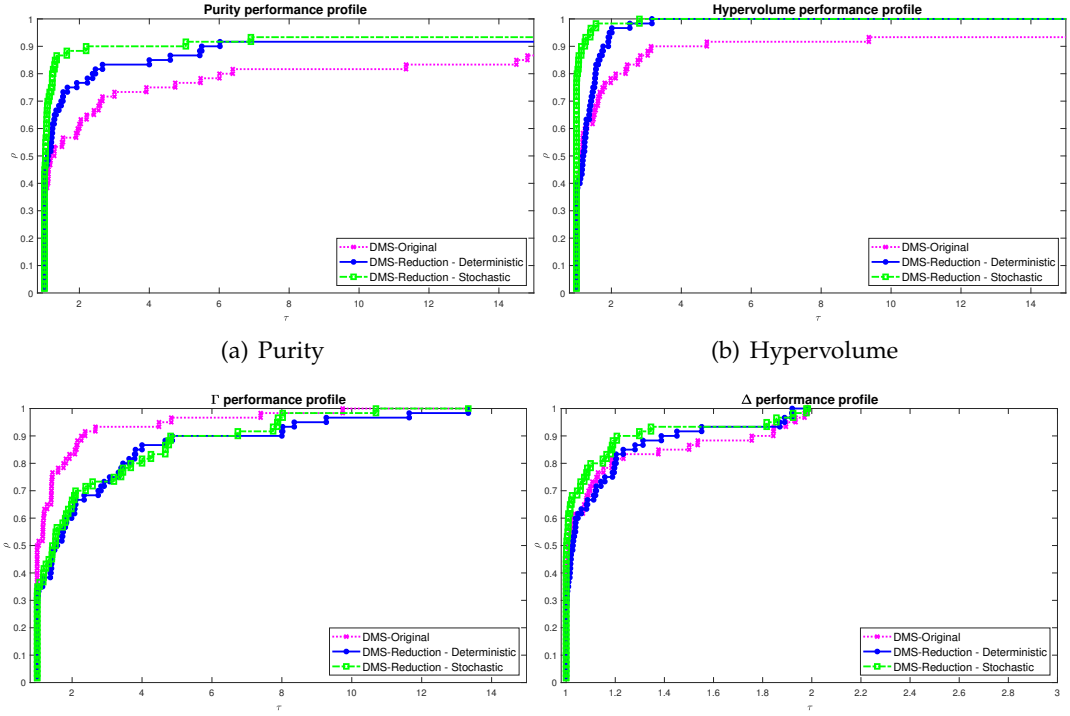
Figure 5.3 presents performance profiles for Purity (a), Hypervolume (b), and Spread, both in its Γ-(c) and Δ-(d) forms, with a maximum budget of 5000 function evaluations. The primary objective here is to discern the most efficient solver among the three versions of DMS using a simple decrease.

With exception of the Γ metric, DMS-Reduction-Stochastic presents the best performance for three additional metrics, both in terms of efficiency and robustness. DMS is the solver with the best performance for Γ.

We also tested the use of sufficient decrease, as a globalization strategy. For that, following the approach suggested in [171] for the single-objective case, we considered the forcing function $\rho(\alpha_k) = c\alpha_k^2 \|P_k^\top d_k\|^2$, with $c = 10^{-9}$. Figure 5.4 depicts the performance of the DMS-Original (using simple decrease) against the previous versions described for DMS-Reduction, but now using sufficient decrease as globalization strategy.

Analyzing Figure 5.4(a), the performance of DMS-Reduction stochastic and deterministic versions based on the Purity metric is similar in terms of efficiency. However, the stochastic version is clearly more robust. It is also clear that the stochastic version of DMS-Reduction is the most efficient algorithm with respect to the hypervolume metric. In terms of the Spread, the solvers present similar performance, with a slight advantage of the original DMS for Γ metric.

In Figure 5.5, we present a comparison between the two best versions of DMS-Reduction, namely the stochastic versions, one with simple decrease and the other utilizing a sufficient decrease approach. The results are really close.



(a) Purity (b) Hypervolume

(c) Spread Gamma (d) Spread Delta

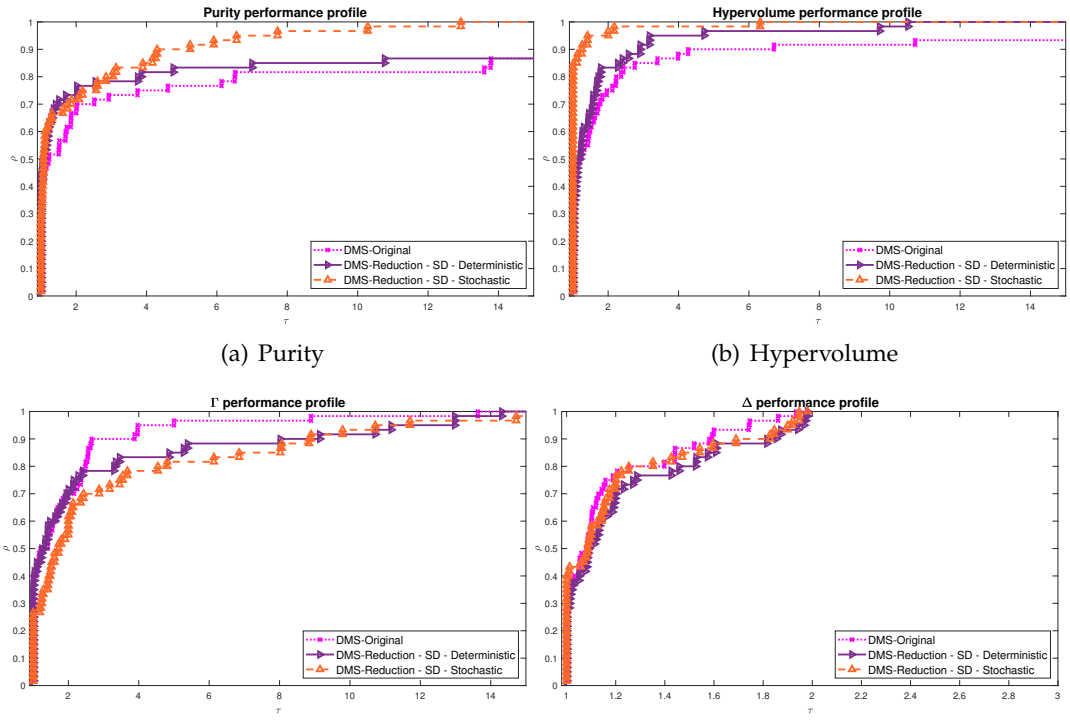
Figure 5.3: Comparing the DMS original with DMS-Reduction (Deterministic and Stochastic versions) based on performance profiles for 5000 function evaluations.

Taking into consideration all the previous results, we can confidently assert that DMS-Reduction (using both globalization approaches) offers distinct advantages over the original DMS version.

In the subsequent segment of our numerical experiments, we aim to thoroughly assess the performance of DMS-Reduction, both in its stochastic and deterministic variants, concerning solvers specifically tailored for addressing many-objective optimization problems. To accomplish it, we have curated a list of solvers from the specialized literature, which is provided below:

- NSGA-III – Nondominated Sorting Genetic Algorithm [71];
- KnEA – Knee point driven Evolutionary Algorithm [187];
- MOEA/DD – Many-objective evolutionary algorithm based on dominance and decomposition [132];
- MOMBI-II – Many-objective metaheuristic based on the R2 indicator II [99];
- GrEA – Grid-Based Evolutionary Algorithm [186].

All the many-objective evolutionary algorithms tested correspond to implementations in the MATLAB-based platform PlatEMO [181], freely available at `https://github.com/BIMK/PlatEMO`. For the next numerical experiments, we allowed a maximum of 5000 and



(a) Purity (b) Hypervolume

(c) Spread Gamma (d) Spread Delta

Figure 5.4: Comparing the DMS original with DMS-Reduction (Deterministic and Stochastic versions using a sufficient decrease approach) based on performance profiles for 5000 function evaluations.

20000 function evaluations for all the solvers. The largest computational budget is justified by the need to allow genetic/evolutionary algorithms to evolve, which typically requires larger budgets of function evaluations.

In Figure 5.6, performance profiles are depicted for the usual metrics and the two computational budgets. DMS-Reduction uses simple decrease to accept new nondominated points.

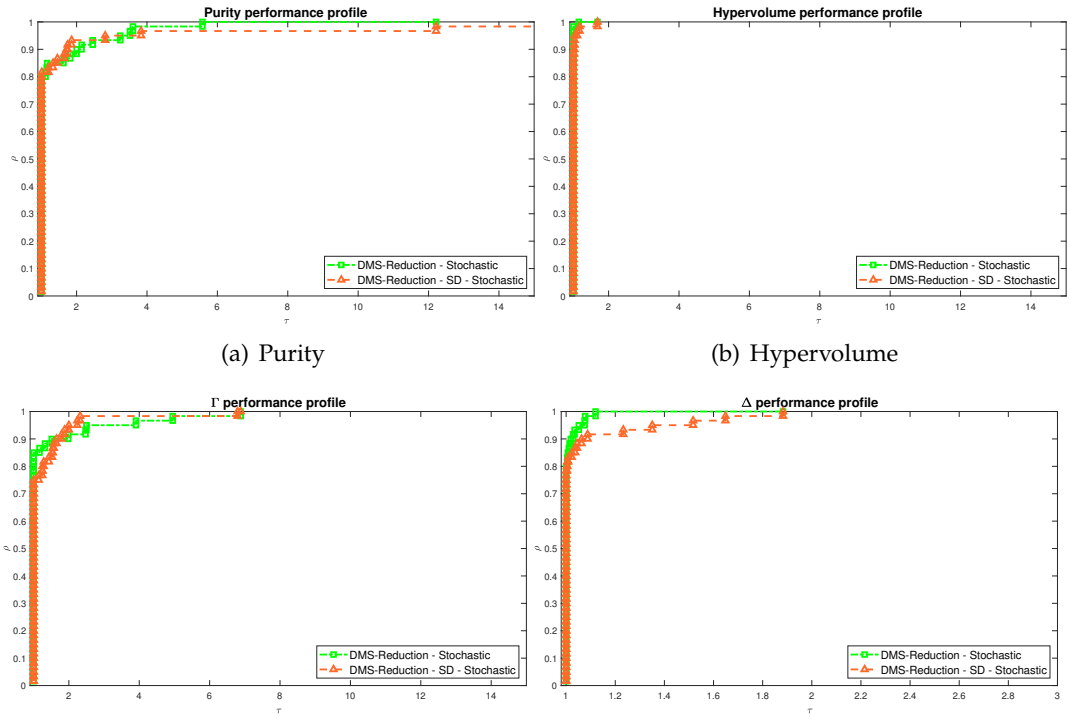
Analyzing hypervolume, DMS-Reduction and GrEA present the best results in terms of efficiency, for the smaller computational budget. Allowing the budget to increase, GrEA and DMS-Reduction-Stochastic are now the top competitors.

However, considering purity, the good performance of GrEA only appears for the larger budget of function evaluations. DMS-Reduction, both for the deterministic and stochastic versions, as well as MOMBI-II, present a good computational performance.

The results of all solvers are very close for Γ, when the smaller computational budget is considered. For the budget of 20000 function evaluations, solvers can be split into groups. MOMBI-II and MOEA/DD constitute the group with worst performance.

Regarding the uniformity of the approximation to the Pareto fronts, again solvers can be grouped into two sets. This time, DMS variants present the worst performance.

Figure 5.7 displays the performance profiles for Purity (a) and (b), the hypervolume metric (c) and (d), Spread Γ (e) and (f), and Spread Δ (g) and (h), for both 5000 and 20000



(a) Purity (b) Hypervolume

(c) Spread Gamma (d) Spread Delta

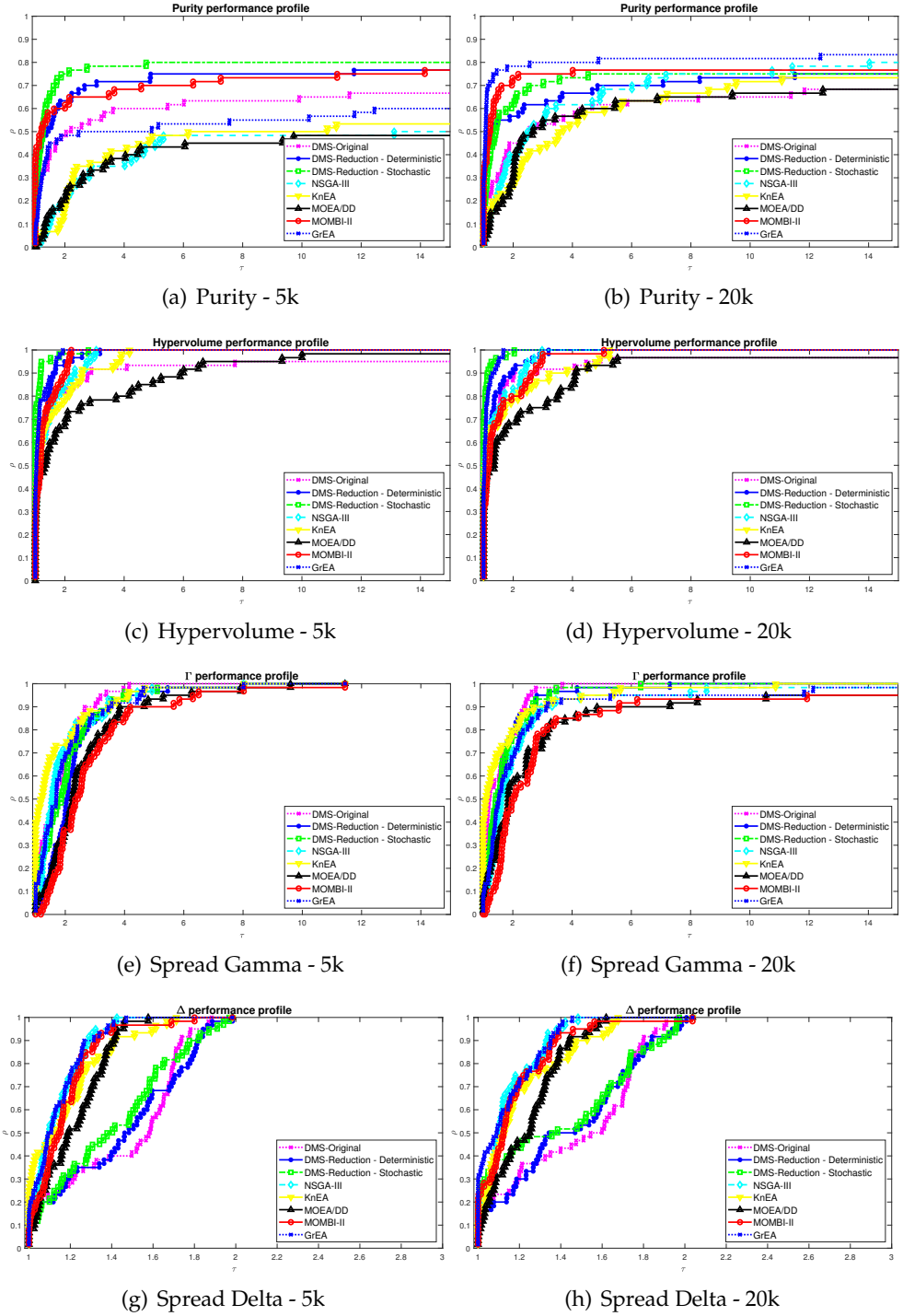
Figure 5.5: Comparing DMS-Reduction-Stochastic based on performance profiles for 5000 function evaluations, using simple and sufficient decrease approaches.

function evaluations. This time, DMS-Reduction employs a sufficient decrease criterion to accept new nondominated points. The conclusions are identical to the ones obtained with the results reported in Figure 5.6.

Gould and Scott [105] demonstrated that when performance profiles are employed to compare more than two solvers, the clarity of the performance profile can diminish significantly. As the number of solvers increases, the profile may become cluttered and more challenging to interpret, making it difficult to discern meaningful differences in solvers performance. Therefore, we have decided to compare only the four most promising solvers (DMS-Reduction, Stochastic and Deterministic versions, MOMBI-II, and GrEA). Results are depicted in Appendix B, in Figures B.1 and B.3, for 5000 function evaluations, and Figures B.2 and B.4, for 20000 function evaluations. In Figures B.1 and B.2, DMS-Reduction employs a simple decrease to accept new nondominated points, while in Figures B.3 and B.4, DMS-Reduction requires sufficient decrease.

In Figure B.1, we observe that the most promising solver is the stochastic version of DMS-Reduction, now considering simple decrease, which exhibits a good performance for all the metrics considered, with exception of Δ. This good performance is kept when the computational budget increases for 20000 function evaluations (see Figure B.2).

Turning to the consideration of sufficient decrease, Figure B.3 illustrates that the performance of the stochastic version declines when constrained to a budget of 5000 function evaluations. Even so, it can still be qualified as a competitive solver. When



(a) Purity - 5k

(b) Purity - 20k

(c) Hypervolume - 5k

(d) Hypervolume - 20k

(e) Spread Gamma - 5k

(f) Spread Gamma - 20k

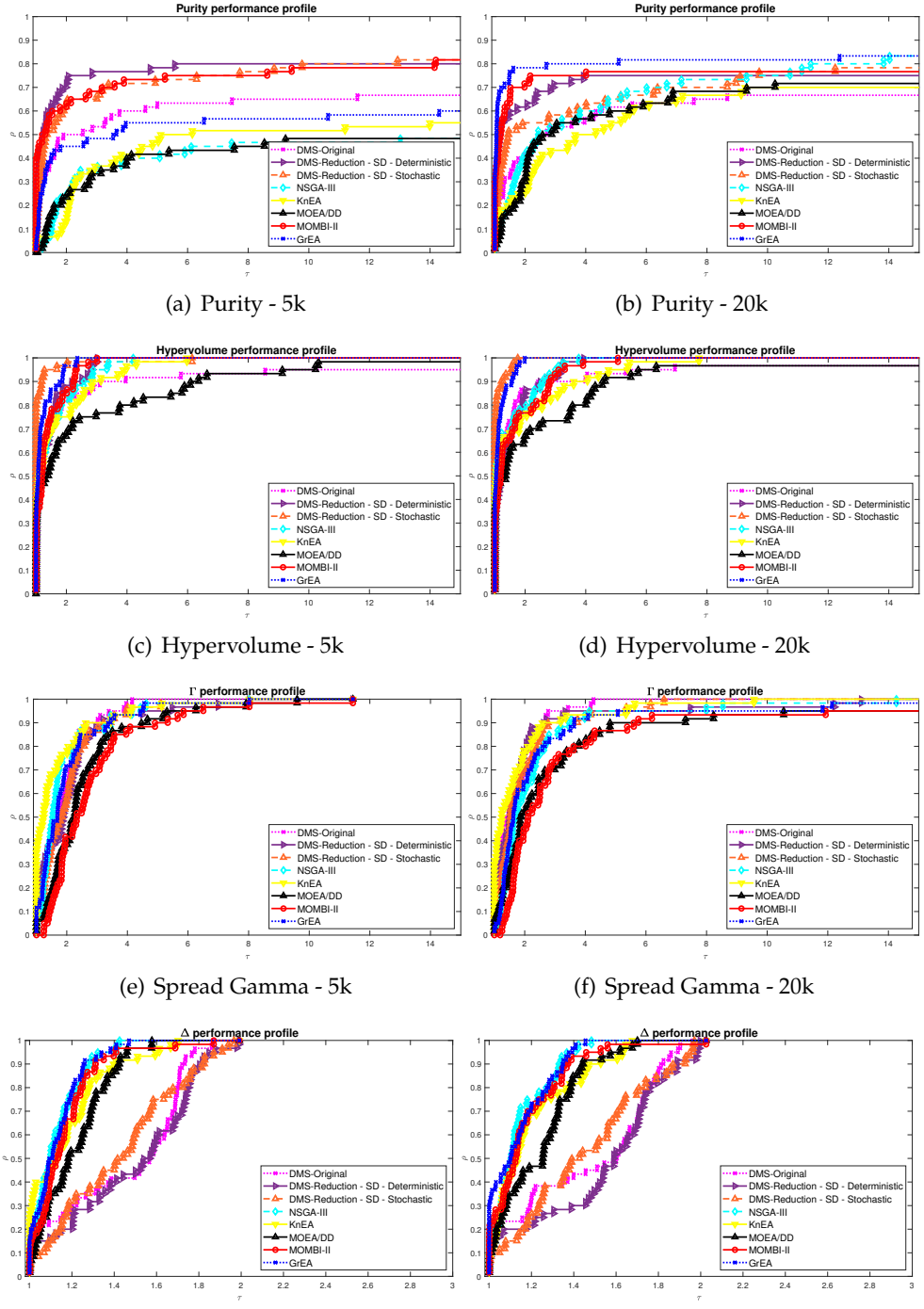
(g) Spread Delta - 5k

(h) Spread Delta - 20k

Figure 5.6: Comparing DMS-Reduction (Stochastic and Deterministic versions, using simple decrease) against DMS, NSGA-III, KnEA, MOEA/DD, MOMBI-II, and GrEA based on performance profiles for 5000 and 20000 function evaluations.

providing a budget of 20000 function evaluations, the performance declines, but it is still competitive in terms of hypervolume and the spread Γ.

Analyzing the experimental results, it becomes evident that the preliminary version of



(a) Purity - 5k (b) Purity - 20k

(c) Hypervolume - 5k (d) Hypervolume - 20k

(e) Spread Gamma - 5k (f) Spread Gamma - 20k

(g) Spread Delta - 5k (h) Spread Delta - 20k

Figure 5.7: Comparing DMS-Reduction (Stochastic and Deterministic versions, using sufficient decrease) against DMS, NSGA-III, KnEA, MOEA/DD, MOMBI-II, and GrEA, based on performance profiles for 5000 and 20000 function evaluations.

DMS-Reduction shows potential to be a competitive solver, exhibiting commendable numerical performance.

5.3.2 A chemical engineering application

In this section, we consider the real chemical engineering problem, related to the production of styrene, as described in [9, 17].

The production of styrene comprises four distinct stages: reactant preparation, which involves pressure increase and evaporation; catalytic reactions; styrene recovery through the first distillation; and benzene recovery through the second distillation. In this second distillation phase, unreacted ethylbenzene is recycled as an initial reactant in the ongoing process. All these procedures can be seen in Figure 5.8.

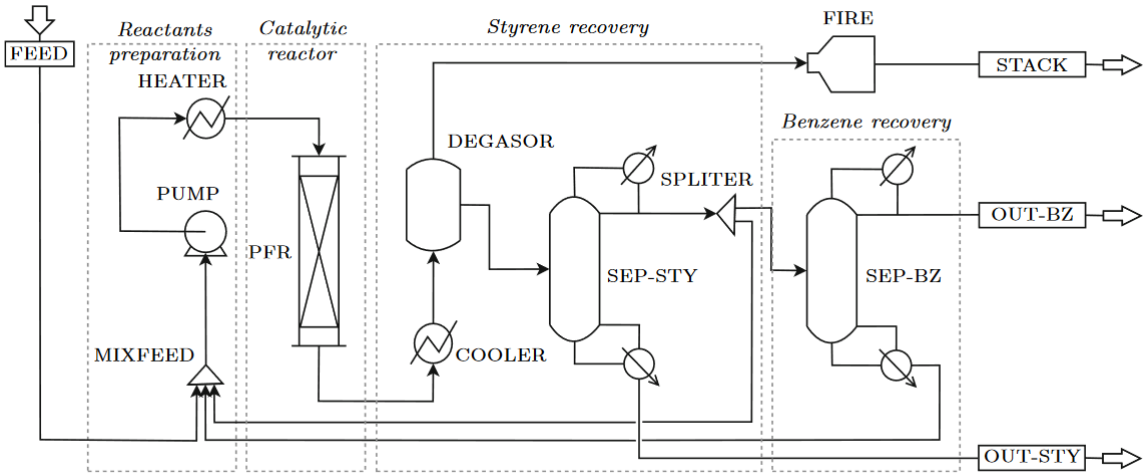


Figure 5.8: Flowsheet of the styrene production process (figure extracted from Audet et al. [9]).

The process can be modeled as a three-objective problem, i.e., $m = 3$, where it is intended to maximize the net present value associated with the production (f_1), the purity of the produced styrene (f_2), and the overall ethylbenzene conversion into styrene (f_3). The problem has 8 variables subject to bounds, five industrial constraints that depend on the block structure and environmental regulations, and four constraints related to payout time, cash flow, investment, and annual costs. More details can be found in [17].

As recycling loops, such as the one involving ethylbenzene, are present, the entire process necessitates simulation until a conclusive result is obtained. Developed in C_{++} and available at `https://github.com/bbopt/styrene`, the chemical process simulator adheres to the Sequential Modular Simulation (SMS) paradigm, with each full simulation corresponding to a function evaluation. The computational time required for evaluating a function value varies, averaging around 1 second when the code successfully evaluates a given point. There could be cases in which a point satisfies all the constraints, but the code fails in returning the corresponding function value. In this case, the simulation is usually faster, requiring an average of 0.01 seconds.

Table 5.2 presents the values for the four different multiobjective metrics for the solvers

DMS [64], DMS with a search step based in quadratic polynomial models, namely BoostDMS [37], a parallel version of BoostDMS, namely Parallel BoostDMS [179], and the stochastic version of DMS-Reduction employing a simple decrease to accept new nondominated points, as described in Section 5.3.1. The number of points in the approximation generated for the Pareto front and the total number of function evaluations performed by each solver are also reported.

Solver	**Purity**	**Hypervolume**	**Gamma**	**Delta**	$\|L\|$	**# Function Evaluations**
DMS	0%	0.0001	7.96e+6	1.0517	7	6741
BoostDMS	0%	0.0989	4.34e+6	1.1228	26	20000
Parallel BoostDMS	0%	0.1321	3.32e+6	1.2266	66	20000
DMS-Reduction	100%	0.9881	7.27e+6	1.1227	16	7995

Table 5.2: Metrics associated with the solution of the chemical engineering problem, computed by solvers DMS (original, with search step, and parallel versions) and Stochastic version of DMS-Reduction employing simple decrease.

The points in the final solution computed by DMS (original, with search step, and parallel versions) are all dominated by the approximation to the Pareto front computed by DMS-Reduction. It is also clear the gain in volume of the dominated region obtained with DMS-Reduction. Figure 5.9 depicts the approximations of the Pareto front generated by each one of the four solvers considered.

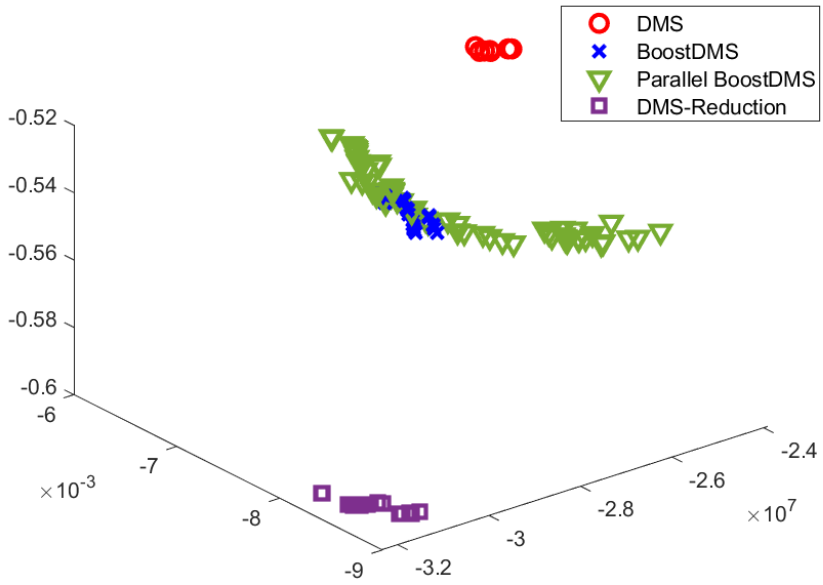


Figure 5.9: Approximations to the Pareto front of the styrene production problem generated by DMS, BoostDMS, Parallel BoostDMS, and the stochastic version of DMS-Reduction, employing simple decrease.

As depicted in Figure 5.9, the original and enhanced versions of DMS occupy distinct zones, with DMS-Reduction emerging as the superior solver among the four versions.

As we have seen, DMS-Reduction is a competitive solver, by comparison with DMS and its variants. However, similarly to the original version, constraints are addressed with an extreme barrier approach only evaluating feasible points. It is worthwhile to access the performance of DMS-Reduction, by comparing it with solvers that consider specific strategies to address constrained problems. For this purpose, we consider DFMO, proposed in [134], and DMultiMADS-PB, proposed in [29]. The DFMO code is written in `Fortran90` and can be found at the URL `http://www.dis.uniroma1.it/~lucidi/DFL`, while the DMultiMADS-PB code, implemented in `Julia`, is available at `https://github.com/bbopt/DMultiMADSPB`. Both the DFMO and DMultiMADS-PB algorithms can handle nonlinear constrained multiobjective problems, penalizing nonlinear constraints with an exact merit function and resourcing to a progressive barrier approach, respectively.

Table 5.3 reports the results for the four metrics (Purity, Hypervolume, Spreads Γ and Δ). As observed, DMS-Reduction emerged as the solver with the smallest number of points in the approximation to the Pareto front. However, it distinguishes itself by presenting the highest percentage of nondominated points (87%). Due to its generation of a limited number of points, and the concentration of these points in a single area, while others are well spread, DMS-Reduction consequently exhibits the lowest hypervolume value.

Solver	**Purity**	**Hypervolume**	**Gamma**	**Delta**	$\lvert L \rvert$	**# Function Evaluations**
DMS-Reduction	87.5%	0.3675	13.8e+6	1.2073	16	7995
DFMO	46.7%	0.7033	5.11e+6	1.6886	122	20000
DMultiMADS-PB	58.4%	0.7149	5.20e+6	1.7989	149	20000

Table 5.3: Metrics associated with the solution of the chemical engineering problem, computed by solvers DMS-Reduction, DFMO, and DMultiMADS-PB.

Figure 5.10 depicts the corresponding approximations to the Pareto front, for each one of the solvers considered.

Considering the crude strategy used by DMS-Reduction to address constraints, the results once again indicate the potential of the solver, also for constrained optimization, particularly if special strategies, such as the one of Chapter 4, are considered.

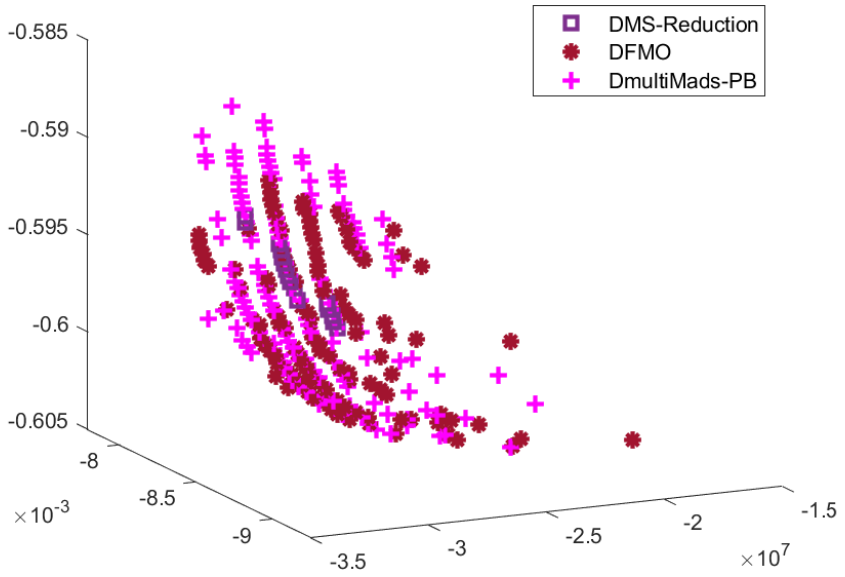


Figure 5.10: Approximations to the Pareto front of the styrene production problem generated by DMS-Reduction, DFMO, and DMultiMADS-PB.

6

Logarithmic Barrier in Direct Search

In this chapter, we consider the use of a logarithmic barrier approach in direct search methods to address both single and multiobjective derivative-free constrained optimization problems. In Section 6.1, we apply the logarithmic barrier method within a single-objective direct search framework. This initial analysis serves as a litmus test to determine whether this strategy yields promising results, and is currently submitted for publication in an international journal [38]. Since the single-objective case can be regarded as a particular instance of the multiobjective framework (when $m = 1$), any lackluster performance in this setting would suggest that extending the approach to the multiobjective case might be unfruitful. Consequently, only after confirming the good performance of the logarithmic barrier method in this simplified scenario we will proceed, in Section 6.2, to adapt and generalize the methodology for tackling multiobjective derivative-free constrained optimization problems.

6.1 Single-objective case

Consider the single-objective derivative-free optimization problem with general constraints (linear and nonlinear), defined by:

$$\begin{aligned} \min\ & f(\boldsymbol{x}) \\ \text{s.t. }\ & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \\ & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \\ & \boldsymbol{x} \in X, \end{aligned} \tag{6.1}$$

where $f : X \to \mathbb{R}$, $g_i : X \to \mathbb{R}$, $i = 1, \ldots, p$, $h_j : X \to \mathbb{R}$, $j = 1, \ldots, q$, are continuously differentiable in X, and $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$, with $\boldsymbol{A} \in \mathbb{R}^{r \times n}$ and $\boldsymbol{b} \in \mathbb{R}^r$. Although, we assume that the derivatives of functions f, g_i, and h_j cannot be either calculated or explicitly approximated.

This section is devoted to introducing a new directional direct search algorithm based on the sequential minimization of an adequate merit function, to solve nonlinearly constrained derivative-free optimization problems. We start by detailing the proposed algorithmic structure, then analyze the role of the linear constraints and their relationship

with the choice of directions to be explored in the poll step, and finally stating the main assumptions required for the theoretical analysis, while also establishing some preliminary results.

The strategy proposed to solve Problem (6.1) builds upon the original idea of [39], where a merit function is defined which guarantees strict feasibility of general inequality constraints by means of logarithmic barrier penalty terms. Equality constraints are treated by exterior penalty terms. The weakness of such an approach is that one needs an initial guess which strictly satisfies all the inequality constraints, otherwise the merit function would not be defined, being the value of infinite assigned to it. We use this requirement to partitioning the original set of nonlinear inequality constraints into two subsets, such that feasibility of the merit function is always guaranteed for the initialization. To this aim, given an initial point x_0, partition the index set of the nonlinear inequality constraints, namely $\{1, 2, \ldots, p\}$, into the following two index sets

$$\mathcal{G}_{\texttt{log}} = \{i \in \{1, \ldots, p\} \mid g_i(x_0) < 0\} \quad \text{and} \quad \mathcal{G}_{\texttt{ext}} = \{i \in \{1, \ldots, p\} \mid g_i(x_0) \geq 0\}.$$

Note that $\mathcal{G}_{\texttt{log}}$ and $\mathcal{G}_{\texttt{ext}}$ depend only on the initial point x_0 and are indeed a partition of the index set of nonlinear inequality constraints, i.e. $\mathcal{G}_{\texttt{log}} \cup \mathcal{G}_{\texttt{ext}} = \{1, \ldots, p\}$ and $\mathcal{G}_{\texttt{log}} \cap \mathcal{G}_{\texttt{ext}} = \emptyset$.

The inequalities $g_i(x) \leq 0$ with $i \in \mathcal{G}_{\texttt{log}}$ will be treated with a logarithmic barrier approach, whereas the ones with $i \in \mathcal{G}_{\texttt{ext}}$ will be aggregated into an exterior penalization term along with the equality constraints. Using the index sets $\mathcal{G}_{\texttt{log}}$ and $\mathcal{G}_{\texttt{ext}}$, we can define

$$\begin{aligned} \Omega_{\texttt{log}} &= \{x \in \mathbb{R}^n \mid g_i(x) \leq 0,\ i \in \mathcal{G}_{\texttt{log}}\}, \\ \Omega_{\texttt{ext}} &= \{x \in \mathbb{R}^n \mid g_i(x) \leq 0,\ i \in \mathcal{G}_{\texttt{ext}} \text{ and } h_j(x) = 0,\ j = 1, \ldots, q\}. \end{aligned}$$

Thus, the feasible region $\mathcal{F}$ of Problem (6.1) is given by $\mathcal{F} = X \cap \Omega_{\texttt{log}} \cap \Omega_{\texttt{ext}}$. Note that, if all inequality constraints are violated by the initial guess, then $\mathcal{G}_{\texttt{log}} = \emptyset$ and $\mathcal{G}_{\texttt{ext}} = \{1, \ldots, p\}$. In this case, $\Omega_{\texttt{log}} = \mathring{\Omega}_{\texttt{log}} = \mathbb{R}^n$. However, for the sake of clarity, we assume that $\mathcal{G}_{\texttt{log}} \neq \emptyset$.

We are now ready to introduce the merit function proposed in this section, which is defined by:

$$Z(x; \rho) = \begin{cases} f(x) - \rho \sum_{i \in \mathcal{G}_{\texttt{log}}} \log(-g_i(x)) \\ \qquad + \frac{1}{\rho^{\nu-1}} \sum_{i \in \mathcal{G}_{\texttt{ext}}} \left(\max\{g_i(x), 0\}\right)^{\nu} & \text{if } x \in X \text{ and } g_i(x) < 0, i \in \mathcal{G}_{\texttt{log}} \\ \qquad + \frac{1}{\rho^{\nu-1}} \sum_{j=1}^{q} |h_j(x)|^{\nu} \\ +\infty & \text{otherwise,} \end{cases} \tag{6.2}$$

where $\rho > 0$ is a *penalty-barrier* parameter and $\nu \in (1, 2]$ plays the role of a *smoothing exponent*. We point out that ρ is the same for the different terms of the merit function, and the range of ν comes from theoretical requirements (see Theorem C.0.4).

Regarding the linear constraints, they are addressed directly during the minimization of $Z(\cdot; \rho)$. Therefore, given a penalty-barrier-parameter ρ_k, the following problem is

considered

$$\begin{aligned} &\min\ Z(\boldsymbol{x};\rho_k) \\ &\text{s.t. } \boldsymbol{x} \in X. \end{aligned} \tag{P_{ρ_k}}$$

Following the classic approach of penalty and interior point methods, the proposed strategy is to generate a sequence of penalty-barrier parameters $\{\rho_k\}$, and sequentially solve the related linearly constrained subproblems. The goal is to obtain a sequence of iterates $\{\boldsymbol{x}_k\}$ that converges to the solution of the original Problem (6.1). Note that, in order for that to happen, one needs the sequence $\{\rho_k\}$ to converge to zero, so that, in the limit, the merit function $Z(\cdot;\rho_k)$ aligns with the objective function $f(\cdot)$ among feasible points.

Taking into account the black-box nature of the functions defining the problem, we propose a directional direct search approach to solve the subproblems (P_{ρ_k}). In particular, as mentioned above, we propose an adaptation of the SID-PSM algorithm, i.e. an implementation of GPS. In the literature, GPS algorithms, when applied to the minimization of a function $\varphi : \mathbb{R}^n \to \mathbb{R}$, accept new points whenever a *simple decrease* of $\varphi(\cdot)$ is obtained, that is, given a current solution $\boldsymbol{x}_k$ and a trial point $\boldsymbol{y}$, whenever $\varphi(\boldsymbol{y}) < \varphi(\boldsymbol{x}_k)$ holds. This forces the method to rely on an implicit mesh to ensure the existence of limit points of the sequence of iterates $\{\boldsymbol{x}_k\}$ (see Section 3.1). In the proposed approach, instead, the acceptance of new points relies on the notion of *sufficient decrease* [69]. The next definition adjusts the concept of *forcing function* (see [121]) and it is used to define the sufficient decrease condition required to accept new points.

Definition 6.1.1. *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 $t \downarrow 0$, and $\xi(t) \to 0$ implies $t \to 0$.*

A classic example of a forcing function is $\xi(t) = \gamma t^2$, with $\gamma > 0$, which is also the most commonly used in practice.
The proposed algorithmic structure is detailed in Algorithm 11.

In the scheme of Algorithm 11, the initial guess $\boldsymbol{x}_0$ is required to satisfy $\boldsymbol{x}_0 \in X \cap \mathring{\Omega}_{\texttt{log}}$, i.e. $x \in X$ and $g_i(\boldsymbol{x}_0) < 0$ for all $i \in \mathcal{G}_{\texttt{log}}$. This ensures that the initial value of the merit function $Z(\boldsymbol{x}_0;\rho_0)$ is finite. It is important to note that $\mathcal{G}_{\texttt{log}}$ and $\mathcal{G}_{\texttt{ext}}$ are determined during the initialization phase and are not modified as the algorithm progresses. The method iterates over four steps. The first three follow the general structure proposed by Audet and Dennis [12] for GPS, whereas the fourth is related to the novel penalty-interior point approach.

Step 1, the *search step*, is very flexible, allowing the use of heuristics, 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 it is detailed in Subsection 6.1.2, the original SID-PSM algorithm uses quadratic polynomial models, which are minimized to generate new trial points. The latter approach is adapted into LOG-DS, but since it is not relevant for establishing convergence properties, for now it is omitted.

Algorithm 11 Logarithmic Barrier in Directional Direct Search - LOG-DS

Initialization

$\mathcal{G}_{\texttt{log}}$ and $\mathcal{G}_{\texttt{ext}}$, $\boldsymbol{x}_0 \in X \cap \mathring{\Omega}_{\texttt{log}}$, $\mathcal{D}$ a set of sets of normalized directions, an initial stepsize $\alpha_0 > 0$, an initial penalty-barrier parameter $\rho_0 > 0$, a smoothing exponent $\nu \in (1, 2]$, a stepsize and penalty-barrier contraction parameters $\theta_\alpha, \theta_\rho \in (0, 1)$, a stepsize expansion parameter $\phi \geq 1$, and $\beta > 1$.

For $k = 0, 1, 2, \ldots$

1. **Search Step:** Compute a finite set of points $\{\boldsymbol{z}_s\}_{s \in S} \in X$. Evaluate Z at each element of $\{\boldsymbol{z}_s\}_{s \in S}$ with the aim of identifying a point $\boldsymbol{z}_s$ that satisfies $Z(\boldsymbol{z}_s; \rho_k) \leq Z(\boldsymbol{x}_k; \rho_k) - \xi(\alpha_k)$. If such a point is found, set $\boldsymbol{x}_{k+1} = \boldsymbol{z}_s$, declare the iteration successful, and skip the poll step.

2. **Poll step:** Select a set $\boldsymbol{D}_k \in \mathcal{D}$. Evaluate Z at the poll points $P_k = \{\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l \mid \boldsymbol{d}_k^l \in \boldsymbol{D}_k\}$. If a polling point $\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l \in X$ is found such that $Z(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l; \rho_k) \leq Z(\boldsymbol{x}_k; \rho_k) - \xi(\alpha_k)$, then stop polling, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l$ and declare the iteration successful. Otherwise, declare the iteration unsuccessful and set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$.

3. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k+1} = \phi \alpha_k$. Otherwise, decrease the stepsize parameter, by setting $\alpha_{k+1} = \theta_\alpha \alpha_k$.

4. **Penalty parameter update:** If the iteration was unsuccessful define $(g_{\min})_k = \min\limits_{i \in \mathcal{G}^{\texttt{log}}} \{|g_i(\boldsymbol{x}_k)|\}$ and if $\alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\min})_k^2\}$, set $\rho_{k+1} = \theta_\rho \rho_k$. Otherwise, set $\rho_{k+1} = \rho_k$.

EndFor

If, during the search step, it is possible to find a point $\boldsymbol{z}_s \in X$ such that, with respect to the current *incumbent* $\boldsymbol{x}_k$, it provides a sufficient decrease in the merit function, i.e., if $Z(\boldsymbol{z}_s; \rho_k) \leq Z(\boldsymbol{x}_k; \rho_k) - \xi(\alpha_k)$, then the algorithm sets $\boldsymbol{x}_{k+1} = \boldsymbol{z}_s$ and proceeds with Step 3. Otherwise, the poll step is invoked.

Step 2, the *poll step*, initializes with the selection of a normalized set of *poll directions* $\boldsymbol{D}_k$. Further considerations on the choice of $\boldsymbol{D}_k$ are discussed latter. If a direction $\boldsymbol{d}_k^l \in \boldsymbol{D}_k$ is found such that the trial point $\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l$ belongs to X and achieves a sufficient decrease in the merit function, then the algorithm sets $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l$ and proceeds with Step 3.

Note that whenever $\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$ one has $Z(\boldsymbol{x}_{k+1}; \rho_k) \leq Z(\boldsymbol{x}_k; \rho_k) - \xi(\alpha_k)$, meaning that, during either the search step or the poll step, the method identifies a better point as candidate to the solution of (P_{ρ_k}). We call such iterations *successful*, and we refer to the index set of successful iterations as K_s. On the other hand, if $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, it means that none of the trial points identified by the poll directions achieves a sufficient decrease in the merit function, i.e. $Z(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k^l; \rho_k) > Z(\boldsymbol{x}_k; \rho_k) - \xi(\alpha_k)$ for all $\boldsymbol{d}_k^l \in \boldsymbol{D}_k$. We call such

iterations *unsuccessful*, and we refer to the index set of unsuccessful iterations as K_u.

Only and at all unsuccessful iterations, Step 4 is invoked. This step concerns the update of the penalty-barrier parameter ρ_k, assessing if (P_{ρ_k}) can be considered as solved. Since providing an exact solution of the problem is, in general, impractical, one has to define a condition to determine if the precision of the solution $\boldsymbol{x}_k$ is acceptable. We propose the following criterion:

$$\alpha_{k+1} \leq \min\{\rho_k^{\beta}, (g_{min})_k^2\},$$

with $\beta > 1$. It is well-known that α_k represents a measure of stationarity for problem (P_{ρ_k}) (see [121]). Hence, having $\alpha_{k+1} \leq \rho_k^{\beta}$ implies that we are not interested in good approximations of the solution of (P_{ρ_k}) for high values of ρ_k. Additionally, $(g_{\min})_k = \min_{i \in \mathcal{G}_{\texttt{log}}}\{|g_i(\boldsymbol{x}_k)|\}$ represents how close we are to the boundary of $\Omega_{\texttt{log}}$. Since the merit function $Z(\boldsymbol{x}; \rho_k)$ takes the value infinite for all $\boldsymbol{x} \notin \mathring{\Omega}_{\texttt{log}}$, by imposing $\alpha_{k+1} \leq (g_{\min})_k^2$ we are trying to force the algorithm to properly explore the interior of $\Omega_{\texttt{log}}$ before moving on to the next subproblem. For further details see [39].

In unconstrained minimization problems, the poll directions should be able to generate $\mathbb{R}^n$ through nonnegative linear combinations, in order to capture the behavior of the objective function (see [61, 112, 113], for further details on the topic). In the presence of linear inequalities, though, the directions must be adapted to the geometry of the set X, identified by the tangent cone. Let us formalize this in the following definition.

Definition 6.1.2 (Active constraints and tangent related sets)**.** *For every $\boldsymbol{x} \in X$, i.e., such that $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, let $\boldsymbol{a}_l^{\top}, l \in \{1, \ldots, r\}$ be the lth row of $\boldsymbol{A}$. We define:*

$$\begin{aligned} I_X(\boldsymbol{x}) &= \{l \in \{1, \ldots, r\} \mid \boldsymbol{a}_l^{\top}\boldsymbol{x} = b_l\} \quad \textit{(set of indices of active constraints)} \\ T_X(\boldsymbol{x}) &= \{\boldsymbol{d} \in \mathbb{R}^n \mid \boldsymbol{a}_l^{\top}\boldsymbol{d} \leq 0,\ l \in I_X(\boldsymbol{x})\} \quad \textit{(tangent cone to } X \textit{ at } \boldsymbol{x}\textit{)} \end{aligned}$$

Since X is convex by definition, for all points $\boldsymbol{x} \in X$ one has that $\boldsymbol{x} + t\boldsymbol{d} \in X$ for all $\boldsymbol{d} \in T_X(\boldsymbol{x})$ and $t > 0$ sufficiently small. Clearly, if $\boldsymbol{x}$ lies in the interior of X, then $T_X(\boldsymbol{x}) = \mathbb{R}^n$.

For any incumbent $\boldsymbol{x}_k$, the tangent cone at $\boldsymbol{x}_k$ might be used to understand the geometry of X at $\boldsymbol{x}_k$ and to build the poll directions accordingly. While this approach might seem reasonable, in general, it is not sufficient. Indeed, the algorithm might generate a sequence of iterates $\{\boldsymbol{x}_k\}$ lying in the interior of X and converging to a point $\bar{\boldsymbol{x}} \in \partial X$. In such case, the tangent cones $T_X(\boldsymbol{x}_k)$ are all equal to $\mathbb{R}^n$, and the algorithm might not be able to sample the space using directions in $T_X(\bar{\boldsymbol{x}}) \subset \mathbb{R}^n$. Thus, given an iterate $\boldsymbol{x}_k \in X$, it is important to be able to capture the geometry of the set X *near* $\boldsymbol{x}_k$. In order to do that, the tangent cone is approximated by the ε-tangent cone [108, 121, 122].

Definition 6.1.3 (ε-Active constraints and tangent related sets)**.**

$$\begin{aligned} I_X(\boldsymbol{x}_k, \varepsilon) &= \{l \in \{1, \ldots, r\} \mid \boldsymbol{a}_l^{\top}\boldsymbol{x}_k \geq b_l - \varepsilon\} \quad \textit{(set of indices of } \varepsilon\textit{-active constraints)} \\ T_X(\boldsymbol{x}_k, \varepsilon) &= \{\boldsymbol{d} \in \mathbb{R}^n \mid \boldsymbol{a}_l^{\top}\boldsymbol{d} \leq 0,\ i \in I_X(\boldsymbol{x}_k, \varepsilon)\} \quad \textit{(}\varepsilon\textit{-tangent cone at } \boldsymbol{x}_k\textit{)} \end{aligned}$$

A relation between the tangent cone and the ε-tangent cone has been established in [138]. Let us recall it in the following proposition.

Proposition 6.1.4. *Let $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ be a sequence of points in X, converging to $\overline{\boldsymbol{x}} \in X$. Then, there exists an $\bar{\varepsilon} > 0$ (depending only on $\overline{\boldsymbol{x}}$) such that for any $\varepsilon \in (0, \bar{\varepsilon}]$ there exists $k_\varepsilon \in \mathbb{N}$ such that*

$$\begin{aligned} I_X(\overline{\boldsymbol{x}}) &= I_X(\boldsymbol{x}_k, \varepsilon) \\ T_X(\overline{\boldsymbol{x}}) &= T_X(\boldsymbol{x}_k, \varepsilon) \end{aligned}$$

for all $k \geq k_\varepsilon$.

Proposition 6.1.4 ensures that, if at each iteration k, the ε-tangent cone of $\boldsymbol{x}_k$ is used to build the set of poll directions $\boldsymbol{D}_k$, then, for small enough $\varepsilon > 0$, the algorithm is able to capture the geometry of X at the limit point $\bar{\boldsymbol{x}}$. The requirements on the sets of poll directions $\boldsymbol{D}_k$ used by the algorithm are formalized in Assumption 6.1.5 (see [135, Assumption 2]).

Assumption 6.1.5. *Let $\{\boldsymbol{x}_k\}$ be a sequence of points in X. The sequence $\{\boldsymbol{D}_k\}$ of poll directions satisfies:*

$$\boldsymbol{D}_k = \{\boldsymbol{d}_k^l \mid \|\boldsymbol{d}_k^l\| = 1, l = 1, \ldots, |\boldsymbol{D}_k|\}$$

and for some $\varepsilon^* > 0$,

$$\text{cone}(\boldsymbol{D}_k \cap T_X(\boldsymbol{x}_k, \varepsilon)) = T_X(\boldsymbol{x}_k, \varepsilon), \quad \forall\, \varepsilon \in (0, \varepsilon^*].$$

Furthermore, $\mathcal{D} = \bigcup_{k=0}^{+\infty} \boldsymbol{D}_k$ is a finite set, and $|\boldsymbol{D}_k|$ is uniformly bounded.

Strategies to conform the search directions to the geometry of the nearby feasible region, i.e. to satisfy Assumption 6.1.5, can be found in [2, 122, 130, 138].

The method we propose is based on the sequential minimization of the merit function $Z(\cdot; \rho_k)$ for a decreasing sequence of penalty-barrier parameters $\{\rho_k\}$. In order for the method to work, an optimal solution has to exist for each Problem (P_{ρ_k}). To guarantee the existence of an optimal solution for a minimization problem, one of the most standard assumptions is the compactness of the lower-level sets of the objective function. In our method, this would correspond to the compactness of the lower level sets $\mathcal{L}_\rho(\alpha) = \{\boldsymbol{x} \in X \mid Z(\boldsymbol{x}; \rho) \leq \alpha\}$ for all $\rho > 0$. Note that the values of $Z(\boldsymbol{x}; \rho)$ are considered only for points lying in X. To this end, we make use of the following assumption, and we prove the desired result in Lemma 6.1.7.

Assumption 6.1.6. *The set $X \cap \Omega_{log}$ is compact and a point $\boldsymbol{x}_0 \in X$ exists such that $g_i(\boldsymbol{x}_0) < 0$ for all $i \in \mathcal{G}_{log}$.*

Lemma 6.1.7. *Let Assumption 6.1.6 hold. Then, for all $\rho > 0$, $\nu \in (1, 2]$, and $\alpha \in \mathbb{R}$ the lower-level set*

$$\mathcal{L}_\rho(\alpha) = \{\boldsymbol{x} \in X \mid\ Z(\boldsymbol{x}; \rho) \leq \alpha\}$$

is compact.

Proof. By definition, $\mathcal{L}_\rho(\alpha)$ is a subset of X, and $Z(\cdot;\rho)$ is coercive with respect to $\Omega_{\texttt{log}}$, i.e. $Z(\boldsymbol{x};\rho) \to +\infty$ as $\boldsymbol{x} \to \partial\Omega_{\texttt{log}}$, thus, $\mathcal{L}_\rho(\alpha) \subseteq X \cap \Omega_{\texttt{log}}$. Since, by Assumption 6.1.6, $X \cap \Omega_{\texttt{log}}$ is compact, then $\mathcal{L}_\rho(\alpha)$ is bounded.

It remains to prove that $\mathcal{L}_\rho(\alpha)$ is closed. To this end, let us show that, for any sequence $\{\boldsymbol{x}_k\} \subset \mathcal{L}_\rho(\alpha)$ such that $\lim_{k\to+\infty} \boldsymbol{x}_k = \bar{\boldsymbol{x}}$, it results $\bar{\boldsymbol{x}} \in \mathcal{L}_\rho(\alpha)$.

Since $\boldsymbol{x}_k \in \mathcal{L}_\rho(\alpha)$ for all k, we have

$$Z(\boldsymbol{x}_k;\rho) \leq \alpha. \tag{6.3}$$

Function $Z(.;\rho)$ is continuous at all $\boldsymbol{x} \in \mathcal{L}_\rho(\alpha)$. Thus, we have

$$\lim_{k\to+\infty} Z(\boldsymbol{x}_k;\rho) = Z(\bar{\boldsymbol{x}};\rho) \leq \alpha,$$

which means that $\bar{\boldsymbol{x}} \in \mathcal{L}_\rho(\alpha)$ and concludes the proof. □

Note that Assumption 6.1.6 arises from the observation that the logarithmic term of $Z(\cdot;\rho)$ might go to $-\infty$ if also $g_i(\cdot) \to -\infty$ for some $i \in \mathcal{G}_{\texttt{log}}$. Since the functions are assumed to be continuous on X, the compactness assumption prevents such situation from happening. Note that, since X is defined by the linear inequalities of the problem, then X also encompasses the possible bounds on the variables. In practical applications, one is usually able to identify some finite bounds wherein the optimal solution should lie. Therefore, the assumption that X is compact can be considered realistic.

We use Lemma 6.1.7 to prove properties that form the basis of the theoretical convergence analysis of Algorithm 11. In particular, it is exploited in the proof of Lemma 6.1.9. Furthermore, it is important to highlight that the results presented in this work do not pertain to the entire sequence generated by the proposed algorithm, but focus on the subset of iterations wherein the penalty-barrier parameter ρ_k is updated, i.e., the sequence of inexact solutions of the linearly constrained Problems (P_{ρ_k}). Following the terminology used in the interior point methods literature (see [27]), we refer to the sequence of inexact solutions generated by Algorithm 11 as a *path following* sequence, and we denote the corresponding index set by K_ρ.

Definition 6.1.8. *Let K_ρ be defined as:*

$$K_\rho = \{k \in \mathbb{N} \mid \ \rho_{k+1} < \rho_k\}.$$

Given the sequence of iterates $\{\boldsymbol{x}_k\}$ produced by Algorithm 11, the subsequence $\{\boldsymbol{x}_k\}_{k\in K_\rho}$ is said to be a path-following subsequence.

Let us recall that Step 4 is invoked only at unsuccessful iterations, i.e., iterations $k \in K_u = \{k \in \mathbb{N} \mid \ x_{k+1} = x_k\}$, so one has $K_\rho \subseteq K_u$. Additionally, according to the instructions of the algorithm, if $k \in K_\rho$ then the following criterion is satisfied:

$$\alpha_{k+1} \leq \{\rho_k^\beta, (g_{min})_k^2\}, \tag{6.4}$$

and $\rho_{k+1} = \theta_\rho \rho_k$, with $\theta_\rho \in (0,1)$. From the updating procedure, it follows that $\{\rho_k\}$ goes to zero only if K_ρ is infinite.

In the next lemma, we prove that, if the set K_ρ is assumed to be finite, then the sequence of stepsizes converges to 0. This technical result is used in Theorem 6.1.10.

Lemma 6.1.9. *Let Assumption 6.1.6 hold. Let $\{\rho_k\}_{k\in\mathbb{N}}$ and $\{\alpha_k\}_{k\in\mathbb{N}}$ be the sequences of penalty-barrier parameters and stepsizes, respectively, generated by Algorithm 11. Assume that K_ρ is finite. Then*

$$\lim_{k\to+\infty} \alpha_k = 0. \tag{6.5}$$

Proof. Without loss of generality, suppose that $K_\rho = \emptyset$. Thus, we have

$$\rho_k = \rho_0, \quad \forall k \in \mathbb{N}.$$

Let us consider the sets of successful and unsuccessful iterations, which we recall to be defined as $K_s = \{k \in \mathbb{N} \mid \boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k\}$ and $K_u = \{k \in \mathbb{N} \mid \boldsymbol{x}_{k+1} = \boldsymbol{x}_k\}$. By the instruction of the algorithm, it follows

$$Z(\boldsymbol{x}_{k+1}; \rho_0) \leq Z(\boldsymbol{x}_k; \rho_0) - \xi(\alpha_k), \quad \forall k \in K_s, \tag{6.6}$$

$$Z(\boldsymbol{x}_{k+1}; \rho_0) = Z(\boldsymbol{x}_k; \rho_0), \quad \forall k \in K_u. \tag{6.7}$$

Hence, the sequence of function values $\{Z(\boldsymbol{x}_k; \rho_0)\}$ is monotonically nonincreasing. By Lemma 6.1.7, $Z(\cdot; \rho_0)$ has compact lower-level sets, thus it is bounded below. Hence,

$$\lim_{k\to+\infty} Z(\boldsymbol{x}_k; \rho_0) = \bar{Z}. \tag{6.8}$$

If K_s is infinite, from (6.8) and (6.6) we get

$$\lim_{\substack{k\to+\infty \\ k\in K_s}} \xi(\alpha_k) = 0.$$

Recalling Definition 6.1.1, we get

$$\lim_{\substack{k\to+\infty \\ k\in K_s}} \alpha_k = 0. \tag{6.9}$$

If K_u is infinite, for every $k \in K_u$, let us define m_k to be the largest index such that $m_k \in K_s$ and $m_k < k$ (the result is immediate if K_s is empty). Then, we can write

$$\alpha_k = \alpha_{m_k} \phi \theta_\alpha^{k-m_k-1}.$$

When $k \to +\infty$, $k \in K_u$, we have that either $m_k \to +\infty$ as well (when K_s is infinite) or $k - m_k - 1 \to +\infty$ (when K_s is finite). Thus, by (6.9) and the fact that $\theta_\alpha \in (0,1)$, we have

$$\lim_{\substack{k\to+\infty \\ k\in K_u}} \alpha_k = \lim_{\substack{k\to+\infty \\ k\in K_u}} \alpha_{m_k} \phi \theta_\alpha^{k-m_k-1} = 0. \tag{6.10}$$

Considering (6.9) and (6.10), the proof is completed. □

As mentioned above, Lemma 6.1.9 is used in the next theorem to show that the index set K_ρ is infinite and, consequently, the sequence of penalty-barrier parameters converges to zero, which is required to ensure that, in the limit, the algorithm is able to solve the original problem. Furthermore, it is possible to establish that the sequence of stepsizes also converges to zero in the general case. Since, as we mentioned above, the stepsize is related to some measure of stationarity, this sets the ground to prove convergence to stationary points.

Theorem 6.1.10. *Let Assumption 6.1.6 hold. Let $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$, $\{\rho_k\}_{k\in\mathbb{N}}$ and $\{\alpha_k\}_{k\in\mathbb{N}}$ be the sequences of iterates, penalty parameters, and stepsizes generated by Algorithm 11, respectively. Then, K_ρ is infinite and*

$$\lim_{k\to+\infty} \rho_k = 0. \tag{6.11}$$

Furthermore, it holds

$$\lim_{\substack{k\to+\infty,\\ k\in K_\rho}} \alpha_k = 0. \tag{6.12}$$

Proof. Let us start by proving that K_ρ is infinite. By contradiction, assume that K_ρ is finite, and without loss of generality, let us assume $K_\rho = \emptyset$. Thus, $\rho_k = \rho_0 > 0$ for all k, and, from Lemma 6.1.9, equation (6.5) holds. Furthermore, by the instructions of Algorithm 11, the sequence $\{Z(\boldsymbol{x}_k;\rho_0)\}$ is monotone non-increasing, i.e. for all k

$$Z(\boldsymbol{x}_{k+1};\rho_0) \leq Z(\boldsymbol{x}_k;\rho_0) \tag{6.13}$$

Let K_s and K_u be the index sets of successful and unsuccessful iterations, respectively. By the instructions of the algorithm, for $k \in K_u$ sufficiently large, since $k \notin K_\rho$ and recalling (6.4), it holds

$$\alpha_{k+1} > \min\{\rho_0^\beta, (g_{\min})_k^2\}.$$

The latter, recalling (6.5), implies

$$\lim_{\substack{k\to+\infty,\\ k\in K_u}} (g_{\min})_k = 0,$$

which, by the definition of $Z(\cdot;\rho_0)$, further implies

$$\lim_{\substack{k\to+\infty,\\ k\in K_u}} Z(\boldsymbol{x}_k;\rho_0) = +\infty.$$

This is a contradiction with (6.13), thus proving that K_ρ is infinite.
Let us now prove (6.11). Let $K_\rho = \{k_1, k_2, \dots, k_j, \dots\}$. Thus, by the instructions of the algorithm, we have

$$\rho_{k_j} = \theta_\rho \rho_{k_{j-1}} = \theta_\rho^j \rho_0.$$

Since K_ρ is infinite, we can take the limit for $j \to +\infty$, and, recalling $\theta_\rho \in (0,1)$, we get

$$\lim_{j\to+\infty} \rho_{k_j} = \lim_{j\to+\infty} \theta_\rho^j \rho_0 = 0,$$

which, noting $\rho_{k+1} \leq \rho_k$ for all k, proves (6.11).
Now, we prove (6.12). By definition of K_ρ, we have for all $k \in K_\rho$

$$\alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\min})_k^2\} \leq \rho_k^\beta.$$

Furthermore, we know that for all iterations $\alpha_{k+1} \geq \theta_\alpha \alpha_k$, so that we can write, for all $k \in K_\rho$,

$$\theta_\alpha \alpha_k \leq \alpha_{k+1} \leq \rho_k^\beta.$$

Then, the proof follows by recalling that $\lim_{k \to +\infty} \rho_k = 0$. □

6.1.1 Convergence analysis

In this subsection, we conduct a theoretical analysis of the convergence properties of Algorithm 11. First, we define the extended Mangasarian–Fromovitz constraint qualification (see Definition 2.1.22), which allows us to derive necessary optimality conditions for Problem (6.1). Then, we present a key technical result, and finally the main finding of the analysis is stated in Theorem 6.1.12.

Before proving the main convergence result, we establish a technical proposition which will be invoked in Theorem 6.1.12. The need for this proposition stems from noting that the trial points corresponding to failures might not strictly satisfy the inequality constraints with indices $i \in \mathcal{G}_{\texttt{log}}$, implying that the merit function value is, by definition, $+\infty$.

Proposition 6.1.11. *Given any $\bar{\rho} > 0$, $\boldsymbol{x} \in X$ such that $g_i(\boldsymbol{x}) < 0$ for all $i \in \mathcal{G}_{\texttt{log}}$, $\boldsymbol{d} \in \mathbb{R}^n$, and $\bar{\alpha} \in \mathbb{R}_+$ such that $\boldsymbol{x} + \bar{\alpha}\boldsymbol{d} \in X$ and*

$$Z(\boldsymbol{x} + \bar{\alpha}\boldsymbol{d}; \bar{\rho}) > Z(\boldsymbol{x}; \bar{\rho}) - \xi(\bar{\alpha}),$$

there exists $\hat{\alpha} \leq \bar{\alpha}$ such that:

$$\begin{aligned}
&\boldsymbol{x} + \alpha\boldsymbol{d} \in X \quad \forall \alpha \in (0, \hat{\alpha}], \\
&\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x} + \alpha\boldsymbol{d}) < 0 \quad \forall\, \alpha \in (0, \hat{\alpha}], \\
&Z(\boldsymbol{x} + \hat{\alpha}\boldsymbol{d}; \bar{\rho}) > Z(\boldsymbol{x}; \bar{\rho}) - \xi(\hat{\alpha}).
\end{aligned}$$

Proof. By definition X is convex, then $\boldsymbol{x}+\alpha\boldsymbol{d} \in X$ for all $\alpha \in [0, \bar{\alpha}]$, thus if $\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x}+\alpha\boldsymbol{d}) < 0$ for all $\alpha \in (0, \bar{\alpha}]$, by setting $\hat{\alpha} = \bar{\alpha}$, the proposition holds.
Let us assume $\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x} + \alpha\boldsymbol{d}) \geq 0$ for some $\alpha \in (0, \bar{\alpha}]$. Since $g_i(\boldsymbol{x}) < 0$ for all $i \in \mathcal{G}_{\texttt{log}}$, it follows that $\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x}) < 0$. Hence, by continuity of $g_i(\cdot)$, there exists a scalar $\tilde{\alpha} < \bar{\alpha}$ such that

$$\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x} + \alpha\boldsymbol{d}) < 0 \;\text{ for all } \alpha \in [0, \tilde{\alpha}),$$

$$\max_{i \in \mathcal{G}_{\texttt{log}}} g_i(\boldsymbol{x} + \tilde{\alpha}\boldsymbol{d}) = 0.$$

Recall that, by definition of $Z(\cdot;\bar{\rho})$, it holds that

$$\lim_{\alpha\to\tilde{\alpha}} Z(\boldsymbol{x}+\alpha\boldsymbol{d};\bar{\rho}) = +\infty.$$

Thus, there exists $\hat{\alpha}\in(0,\tilde{\alpha})$, sufficiently close to $\tilde{\alpha}$, such that $g_i(\boldsymbol{x}+\hat{\alpha}\boldsymbol{d})<0$, for all $i\in\mathcal{G}_{\texttt{log}}$, and $Z(\boldsymbol{x};\bar{\rho}) < Z(\boldsymbol{x}+\hat{\alpha}\boldsymbol{d};\bar{\rho})+\xi(\hat{\alpha})$, which concludes the proof. □

We are now in the condition of stating the main convergence result, stationarity in the sense of Definition 2.1.23.

Theorem 6.1.12. *Let Assumption 6.1.6 hold. Let $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ be the sequence of iterates generated by Algorithm 11 and let K_ρ be a path following sequence. Assume that the sets of directions $\{\boldsymbol{D}_k\}_{k\in\mathbb{N}}$, used by the algorithm, satisfy Assumption 6.1.5 and define $J_k=\{l\in\{1,2,\dots,|\boldsymbol{D}_k|\}\mid \boldsymbol{d}_k^l\in\boldsymbol{D}_k\cap T_X(\boldsymbol{x}_k;\varepsilon)\}$, with $\varepsilon\in(0,\min\{\varepsilon^*,\bar{\varepsilon}\}]$ where $\bar{\varepsilon}$ and ε^* are the constants appearing in Proposition 6.1.4 and Assumption 6.1.5, respectively. Then, any limit point of $\{\boldsymbol{x}_k\}_{k\in K_\rho}$ that satisfies the MFCQ is a stationary point of Problem (6.1).*

Proof. First note that, by Theorem 6.1.10, we have

$$\lim_{\substack{k\to+\infty\\ k\in K_\rho}} \rho_k = 0,$$
$$\lim_{\substack{k\to+\infty\\ k\in K_\rho}} \alpha_k = 0.$$

Now, let $\overline{\boldsymbol{x}}$ be any limit point of $\{\boldsymbol{x}_k\}_{k\in K_\rho}$. Then, there exists a set $K_\rho^{\boldsymbol{x}}\subseteq K_\rho$ such that

$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{\boldsymbol{x}}}} \rho_k = 0,$$
$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{\boldsymbol{x}}}} \alpha_k = 0,$$
$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{\boldsymbol{x}}}} \boldsymbol{x}_k = \overline{\boldsymbol{x}},$$

with $\alpha_{k+1}<\alpha_k$ and $\rho_{k+1}<\rho_k$, for all $k\in K_\rho^{\boldsymbol{x}}$. Recall that $\boldsymbol{D}_k=\{\boldsymbol{d}_k^1,\boldsymbol{d}_k^2,\dots,\boldsymbol{d}_k^{|\boldsymbol{D}_k|}\}$. Then, for all $k\in K_\rho^{\boldsymbol{x}}$ sufficiently large, we know that $\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k^l\in X$ for all $l\in J_k$. For every $l\in J_k$, if $g_i(\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k^l)<0$, for all $i\in\mathcal{G}_{\texttt{log}}$, by the instructions of the algorithm we have

$$Z(\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k^l;\rho_k) > Z(\boldsymbol{x}_k;\rho_k)-\xi(\alpha_k). \tag{6.14}$$

Otherwise, i.e. when an index $i\in\mathcal{G}_{\texttt{log}}$ exists such that $g_i(\boldsymbol{x}_k+\alpha_k\boldsymbol{d}_k^l)\geq 0$, i.e. $Z(\boldsymbol{x}^k+\alpha_k\boldsymbol{d}_k^l;\rho_k)=+\infty$, Proposition 6.1.11 allows us to ensure the existence of a scalar $\hat{\alpha}_k^l\leq\alpha_k$ such that

$$Z(\boldsymbol{x}_k+\hat{\alpha}_k^l\boldsymbol{d}_k^l;\rho_k) > Z(\boldsymbol{x}_k;\rho_k)-\xi(\hat{\alpha}_k^l). \tag{6.15}$$

Applying the mean value theorem to (6.15) (or (6.14) setting $\hat{\alpha}_k^l=\alpha_k$), we can write

$$-\xi(\hat{\alpha}_k^l)\leq Z(\boldsymbol{x}_k+\hat{\alpha}_k^l\boldsymbol{d}_k^l;\rho_k)-Z(\boldsymbol{x}_k;\rho_k)=\hat{\alpha}_k^l\nabla Z(\boldsymbol{y}_k^l;\rho_k)^\top\boldsymbol{d}_k^l, \tag{6.16}$$

for all $k \in K_\rho^{\boldsymbol{x}}$ sufficiently large and all $l \in J_k$, where $\boldsymbol{y}_k^l = \boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l$, with $t_k^l \in (0,1)$. Thus, we have

$$\nabla Z(\boldsymbol{y}_k^l; \rho_k)^\top \boldsymbol{d}_k^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l}, \quad \forall l \in J_k. \tag{6.17}$$

By considering the expression of $Z(\boldsymbol{x}; \rho_k)$, for all $l \in J_k$ and for $k \in K_\rho^{\boldsymbol{x}}$ sufficiently large

$$\nabla Z(\boldsymbol{y}_k^l; \rho_k)^\top \boldsymbol{d}_k^l = \left[\nabla f(\boldsymbol{y}_k^l) + \sum_{i \in \mathcal{G}_{\texttt{log}}} \frac{\rho_k}{-g_i(\boldsymbol{y}_k^l)} \nabla g_i(\boldsymbol{y}_k^l) + \nu \left(\sum_{i \in \mathcal{G}_{\texttt{ext}}} \left(\frac{\max\{g_i(\boldsymbol{y}_k^l), 0\}}{\rho_k} \right)^{\nu-1} \nabla g_i(\boldsymbol{y}_k^l) + \right. \right.$$
$$\left. \left. \sum_{j=1}^{q} \left(\frac{|h_j(\boldsymbol{y}_k^l)|}{\rho_k} \right)^{\nu-1} \nabla h_j(\boldsymbol{y}_k^l) \right) \right]^\top \boldsymbol{d}_k^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l}. \tag{6.18}$$

By Assumption 6.1.5, we can extract a further subset of indices $K_\rho^{\boldsymbol{x},\boldsymbol{D}} \subseteq K_\rho^{\boldsymbol{x}}$ such that the sets of poll directions $\boldsymbol{D}_k$ are equal to a fixed set $\overline{\boldsymbol{D}}$ for all $k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}$. Therefore, we can write

$$\begin{aligned}
&\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}} \rho_k = 0 \\
&\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}} \alpha_k = 0, \\
&\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}} \boldsymbol{x}_k = \overline{\boldsymbol{x}}, \\
&J_k = \overline{J}, \quad \forall\, k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}, \\
&\boldsymbol{d}_k^l = \bar{\boldsymbol{d}}^l, \quad \forall\, l \in \overline{J},\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}, \\
&\overline{\boldsymbol{D}} = \{\bar{\boldsymbol{d}}^l\}_{l \in \overline{J}}.
\end{aligned}$$

When $k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}$ is sufficiently large, for all $l \in \overline{J}$, with $\boldsymbol{y}_k^l = \boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \bar{\boldsymbol{d}}^l$, and $t_k^l \in (0,1)$, since $\hat{\alpha}_k^l \leq \alpha_k$, by Theorem 6.1.10, we have that, $\lim\limits_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}} \boldsymbol{y}_k^l = \overline{\boldsymbol{x}}$.

Let us define the following approximations to the Lagrange multipliers of each constraint:

- for $i = 1, \ldots, p$ set $\lambda_i(\boldsymbol{x}; \rho) = \begin{cases} \dfrac{\rho}{-g_i(\boldsymbol{x})}, & \text{if } i \in \mathcal{G}_{\texttt{log}} \\ \nu \left(\dfrac{\max\{g_i(\boldsymbol{x}), 0\}}{\rho} \right)^{\nu-1}, & \text{if } i \in \mathcal{G}_{\texttt{ext}} \end{cases}$

- for $j = 1, \ldots, q$ set $\mu_j(\boldsymbol{x}; \rho) = \nu \left(\dfrac{|h_j(\boldsymbol{x})|}{\rho} \right)^{\nu-1}$.

The sequences $\{\lambda_i(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}$ and $\{\mu_j(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^{\boldsymbol{x},\boldsymbol{D}}}$, are bounded (see Appendix C). Thus, it is possible to consider $K_\rho^{\boldsymbol{x},\boldsymbol{D},\boldsymbol{\lambda}} \subseteq K_\rho^{\boldsymbol{x},\boldsymbol{D}}$, such that

$$\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D},\boldsymbol{\lambda}}}} \lambda_i(\boldsymbol{x}_k; \rho_k) = \overline{\lambda}_i, \qquad i = 1, \ldots, p \tag{6.19}$$

$$\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{D},\boldsymbol{\lambda}}}} \mu_j(\boldsymbol{x}_k; \rho_k) = \overline{\mu}_j, \qquad j = 1, \ldots, q. \tag{6.20}$$

Note that, by definition of the multiplier functions, we have $\overline{\lambda}_i = 0$ for $i \notin I_+(\overline{x})$.

Multiplying (6.18) by $\rho_k^{\nu-1}$ and taking the limit for $k \to +\infty, k \in K_\rho^{x,D,\lambda}$, recalling $\nu \in (1,2]$, we have that $\rho_k^{\nu-1} \to 0$, so we obtain the following

$$\left(\sum_{i \in \mathcal{G}_{\text{ext}}} \nu \max\{g_i(\overline{x}), 0\}^{\nu-1} \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top \bar{d}^l \geq 0, \quad \forall \bar{d}^l \in \overline{D}.$$

From Proposition 6.1.4 and Assumption 6.1.5, since $\varepsilon \in (0, \min\{\bar{\varepsilon}, \varepsilon^*\}]$, for all $k \in K_\rho^{x,D,\lambda}$ sufficiently large it holds

$$T_X(\overline{x}) = T_X(x_k; \varepsilon) = \text{cone}(D_k \cap T_X(x_k; \varepsilon)) = \text{cone}(\overline{D}).$$

Then, for every $d \in T_X(\overline{x})$, there exist nonnegative numbers β_l such that

$$d = \sum_{l \in \bar{J}} \beta_l \bar{d}^l, \text{with } \bar{d}^l \in \overline{D}. \tag{6.21}$$

Let us recall that, by assumption, $\overline{x}$ satisfies the MFCQ conditions, and let d be the direction satisfying (2.7) in point (b) of the MFCQ conditions. Then we have,

$$\begin{aligned} 0 \leq \quad & \sum_{l \in \bar{J}} \beta_l \left(\sum_{i \in \mathcal{G}_{\text{ext}}} \nu \max\{g_i(\overline{x}), 0\}^{\nu-1} \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top \bar{d}^l = \\ & \left(\sum_{i \in \mathcal{G}_{\text{ext}}} \nu \max\{g_i(\overline{x}), 0\}^{\nu-1} \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top d = \\ & \left(\sum_{i \in I_+(\overline{x}) \cap \mathcal{G}_{\text{ext}}} \nu \max\{g_i(\overline{x}), 0\}^{\nu-1} \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top d, \end{aligned} \tag{6.22}$$

where the last equality follows by considering that $\max\{g_i(\overline{x}), 0\} = 0$, for all $i \in \mathcal{G}_{\text{ext}} \setminus I_+(\overline{x})$.

Again by (2.7), $\nabla g_i(\overline{x})^T d < 0$, for all $i \in I_+(\overline{x})$, and $\nabla h_j(\overline{x})^\top d = 0$, for all j. Then, we get $\max\{g_i(\overline{x}), 0\} = 0$ for all $i \in I_+(\overline{x}) \cap \mathcal{G}_{\text{ext}}$, and we can conclude that $g_i(\overline{x}) \leq 0$ for all $i \in \mathcal{G}_{\text{ext}}$. Furthermore, (6.22) and $g_i(\overline{x}) \leq 0$ for all $i \in \mathcal{G}_{\text{ext}}$, imply

$$\begin{aligned} & \left(\sum_{i \in \mathcal{G}_{\text{ext}}} \nu \max\{g_i(\overline{x}), 0\}^{\nu-1} \nabla g_i(\overline{x}) + \sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top d = \\ & \qquad \left(\sum_{j=1}^{q} \nu |h_j(\overline{x})|^{\nu-1} \nabla h_j(\overline{x}) \right)^\top d \geq 0, \quad \text{for all } d \in T_X(\overline{x}). \end{aligned}$$

Using (2.6), we get $h_j(\overline{x}) = 0$ for all $j = 1, \ldots, q$. Therefore, recalling that $g_i(\overline{x}) \leq 0$ for all $i \in \mathcal{G}_{\text{log}}$, we obtain that point $\overline{x}$ is feasible.

By simple manipulations, inequality (6.18) can be rewritten as

$$\begin{aligned}\Bigg(&\nabla f(\boldsymbol{y}_k^l) + \sum_{i=1}^{p} \nabla g_i(\boldsymbol{y}_k^l)\lambda_i(\boldsymbol{x}_k;\rho_k) \\ &+ \sum_{i=1}^{p} \nabla g_i(\boldsymbol{y}_k^l)(\lambda_i(\boldsymbol{y}_k^l;\rho_k) - \lambda_i(\boldsymbol{x}_k;\rho_k)) + \sum_{j=1}^{q} \nabla h_j(\boldsymbol{y}_k^l)\mu_j(\boldsymbol{x}_k;\rho_k) \\ &+ \sum_{j=1}^{q} \nabla h_j(\boldsymbol{y}_k^l)(\mu_j(\boldsymbol{y}_k^l;\rho_k) - \mu_j(\boldsymbol{x}_k;\rho_k))\Bigg)^{\top} \bar{\boldsymbol{d}}^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l}, \quad \forall l \in \bar{J}\end{aligned} \tag{6.23}$$

Taking limits for $k \to +\infty$, $k \in K_\rho^{\boldsymbol{x},\boldsymbol{D},\boldsymbol{\lambda}}$ and considering (C.14) and (C.15) in Appendix C, we get:

$$\left(\nabla f(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \nabla g_i(\overline{\boldsymbol{x}})\overline{\lambda}_i + \sum_{j=1}^{q} \nabla h_j(\overline{\boldsymbol{x}})\overline{\mu}_j\right)^{\top} \bar{\boldsymbol{d}}^l \geq 0, \quad \forall l \in \bar{J}. \tag{6.24}$$

Again, by (6.21) and (6.24), we have, for all $\boldsymbol{d} \in T_X(\overline{\boldsymbol{x}})$,

$$\begin{aligned}&\left(\nabla f(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \nabla g_i(\overline{\boldsymbol{x}})\overline{\lambda}_i + \sum_{j=1}^{q} \nabla h_j(\overline{\boldsymbol{x}})\overline{\mu}_j\right)^{\top} \boldsymbol{d} = \\ &\sum_{l\in\bar{J}} \beta_l \left(\nabla f(\overline{\boldsymbol{x}}) + \sum_{i=1}^{p} \nabla g_i(\overline{\boldsymbol{x}})\overline{\lambda}_i + \sum_{j=1}^{q} \nabla h_j(\overline{\boldsymbol{x}})\overline{\mu}_j\right)^{\top} \bar{\boldsymbol{d}}^l \geq 0.\end{aligned}$$

Since $\overline{\boldsymbol{x}}$ is feasible and considering that, by definition of $\overline{\lambda}_i$ for $i \in I_+(\overline{\boldsymbol{x}})$, we have $\overline{\lambda}_i\, g_i(\overline{\boldsymbol{x}}) = 0$, for all $i = 1, \ldots, p$, $\overline{\boldsymbol{x}}$ is a KKT point for problem (6.1), thus concluding the proof. □

6.1.2 Implementation details of LOG-DS

In this subsection, we describe a practical implementation for LOG-DS, based on the original implementation of SID-PSM [65, 66].

LOG-DS enhances SID-PSM with the ability of handling general constraints using a penalty-interior point method. Thus, the original structure and algorithmic options of SID-PSM implementation were considered. In this subsection, we provide a general overview of the main features of SID-PSM, highlighting the differences with LOG-DS. For more details on SID-PSM, the original references [65, 66] 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 the violation of nonlinear constraints. The merit function also replaces the original objective function throughout the different algorithmic steps. So, during the Search Step, quadratic polynomial models are built for both the objective and constraints functions

and they are analytically combined to produce a model of the merit function. The model is minimized inside a ball with a radius length directly related to the stepsize 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 [65] to compute the model coefficients.

The original SID-PSM only requires a simple decrease for the acceptance of new points, whereas in LOG-DS points are accepted if they satisfy the sufficient decrease condition

$$Z(\boldsymbol{x}_{k+1}; \rho_k) \leq Z(\boldsymbol{x}_k; \rho_k) - \gamma \alpha_k^2,$$

with $\gamma = 10^{-9}$.

The algorithm proceeds with an *opportunistic* Poll Step, accepting the first tentative point satisfying the sufficient decrease condition. Before initiating the polling procedure, previously evaluated points are again used to build a *simplex gradient* [34], which is used as an ascent indicator. Poll directions are reordered according to the largest angle made with this ascent indicator (see [66]). In LOG-DS, the simplex gradient is built on the merit function values, while the original SID-PSM exploits the values of the objective function.

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 $\boldsymbol{x}_0 \in X$:

$$\mathcal{G}_{\texttt{log}} = \{i \in \{1, \ldots, p\} \mid g_i(\boldsymbol{x}_0) < 0\} \text{ and } \mathcal{G}_{\texttt{ext}} = \{i \in \{1, \ldots, p\} \mid g_i(\boldsymbol{x}_0) \geq 0\}.$$

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 $\rho^{\texttt{ext}}$ as $\rho^{\nu-1}$ in equation (6.2). Therefore, we can write the merit function as

$$Z(\boldsymbol{x}; \rho_k) = \begin{cases} f(\boldsymbol{x}) - \rho_k^{\texttt{log}} \sum_{i \in \mathcal{G}_{\texttt{log}}} \log(-g_i(\boldsymbol{x})) \\ \qquad + \frac{1}{\rho_k^{\texttt{ext}}} \sum_{i \in \mathcal{G}_{\texttt{ext}}} \left(\max\{g_i(\boldsymbol{x}), 0\}\right)^{\nu} & \text{if } \boldsymbol{x} \in X \text{ and } g_i(\boldsymbol{x}) < 0, i \in \mathcal{G}_{\texttt{log}} \\ \qquad + \frac{1}{\rho_k^{\texttt{ext}}} \sum_{j=1}^{q} |h_j(\boldsymbol{x})|^{\nu} \\ +\infty & \text{otherwise,} \end{cases} \tag{6.25}$$

The penalty parameters are updated only at unsuccessful iterations, considering two different criteria:

$$\alpha_{k+1} \leq \min\{(\rho_k^{\texttt{log}})^{\beta}, (g_{\min})_k^2\}, \text{ for updating } \rho_k^{\texttt{log}}; \tag{6.26}$$

$$\alpha_{k+1} \leq \min\{(\rho_k^{\texttt{log}})^{\beta}, (\rho_k^{\texttt{ext}})^{\beta}, (g_{\min})_k^2\}, \text{ when updating } \rho_k^{\texttt{ext}}. \tag{6.27}$$

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 (6.2).

If inequality (6.26) holds, LOG-DS uses the following rule

$$\rho_{k+1}^{\texttt{log}} = \zeta \rho_k^{\texttt{log}}, \tag{6.28}$$

whereas if inequality (6.27) holds, LOG-DS performs the following update

$$\rho_{k+1}^{\texttt{ext}} = \zeta \rho_k^{\texttt{ext}}, \tag{6.29}$$

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 able to attain feasibility. While scaling the initial exterior penalty parameter with respect to the initial value of the objective function improved our 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 some 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.

6.1.3 Numerical experiments

This subsection is dedicated to presenting 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, when compared against SID-PSM and other state-of-art solvers.

We considered the test set used in [39], part of the CUTEst collection [106]. Specifically, we included problems where the number of variables does not exceed 50 and at least one inequality constraint is strictly satisfied by the initialization provided, ensuring $\boldsymbol{x}_0 \in \mathring{\Omega}_{\texttt{log}}$.

Table 6.1 details the test set, providing the name, the number of variables, n, the number of inequality constraints, p, the number of inequality constraints treated by the logarithmic barrier $\bar{p} = |\mathcal{G}_{\texttt{log}}|$, and the number of nonlinear equalities, q, for each problem. Additional information can be found in [39, 106].

6.1.3.1 Strategies for addressing linear constraints

We start by evaluating the performance of LOG-DS using two distinct approaches for managing linear inequality constraints, other than bounds. The first approach treats each linear inequality constraint (other than bounds) as a general inequality, i.e. addressing it within the proposed penalty-barrier approach. The second approach consists on the

Problem	n	p	$\bar{p}$	q
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

Problem	n	p	$\bar{p}$	q
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 6.1: Test set selected from the CUTEst collection. Parameters n, p, $\bar{p}$, and q 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.

use of an extreme barrier method, adjusting the directions so that Assumption 6.1.5 is satisfied for the linear constraints (including bounds).

It is important to understand the relevance of comparing the two strategies. The logarithmic barrier approach is well-known for handling linear constraints very efficiently, especially in the presence of a large number. Moreover, 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 the coordinate directions, which are known to have good geometry for building linear models. Any other orthonormal basis could be used, though the coordinate directions additionally conform to bound constraints on the variables, so that we can treat them separately from the penalty approach.

The works by Lucidi et al. [138] and Lewis and Torczon [130] propose methods for computing directions conforming to linear inequality constraints but do not consider degeneracy. Abramson et al. [2] provide a detailed algorithm for generating a set directions with the desired property, regardless of whether the constraints are degenerate or not.

In order to use Definition 6.1.3, of ε-active constraints, we assume a preliminary scaling of the constraints. For this purpose, we multiply the lth constraint by $\|\boldsymbol{a}_l\|^{-1}$, since the vectors $\boldsymbol{a}_l$, $l \in I_X(\boldsymbol{x}_k)$, are not null (we have that $\|\boldsymbol{a}_l\| \neq 0$ for all $l \in I_X(\boldsymbol{x}_k)$). Therefore, we consider

$$\overline{\boldsymbol{a}}_l = \frac{\boldsymbol{a}_l}{\|\boldsymbol{a}_l\|}, \quad \overline{b}_l = \frac{b_l}{\|\boldsymbol{a}_l\|}, \quad l \in I_X(\boldsymbol{x}_k). \tag{6.30}$$

The ε-active index set is computed using the matrix $\overline{\boldsymbol{A}}$, a scaled version of the matrix $\boldsymbol{A}$, and the vector $\overline{\boldsymbol{b}}$. Consequently, we have that $\|\overline{\boldsymbol{a}}_l\| = 1$ for all $l \in I_X(\boldsymbol{x}_k)$ and $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} \leq \mathbf{b}\} = \{\boldsymbol{x} \in \mathbb{R}^n \mid \overline{\boldsymbol{A}}\boldsymbol{x} \leq \overline{\mathbf{b}}\}$.

To compute the set of directions $\boldsymbol{D}_k$ that conforms to the geometry of the nearby constraints, we use the algorithm proposed in [2, Alg. 4.4.2]. The latter is divided into two parts: the first constructs the index set corresponding to the ε-active non-redundant constraints, and the second computes the set of directions $\boldsymbol{D}_k$, which include the generators of the cone $T_X(\boldsymbol{x}_k, \varepsilon)$. For identifying a constraint as being ε-active, we considered $\varepsilon = 10^{-5}$. When linear constraints (other than bounds) are included in the merit function, we used $\boldsymbol{D}_k = [\,\mathbf{1} \;\; -\mathbf{1} \quad \mathbb{I}_n \;\; -\mathbb{I}_n\,]$ as the default set of directions, for every k.

Figure 6.1 depicts the performance of LOG-DS using the two strategies described before, considering a maximum number of 2000 function evaluations and a minimum stepsize tolerance equal to 10^{-8}.

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 by itself.

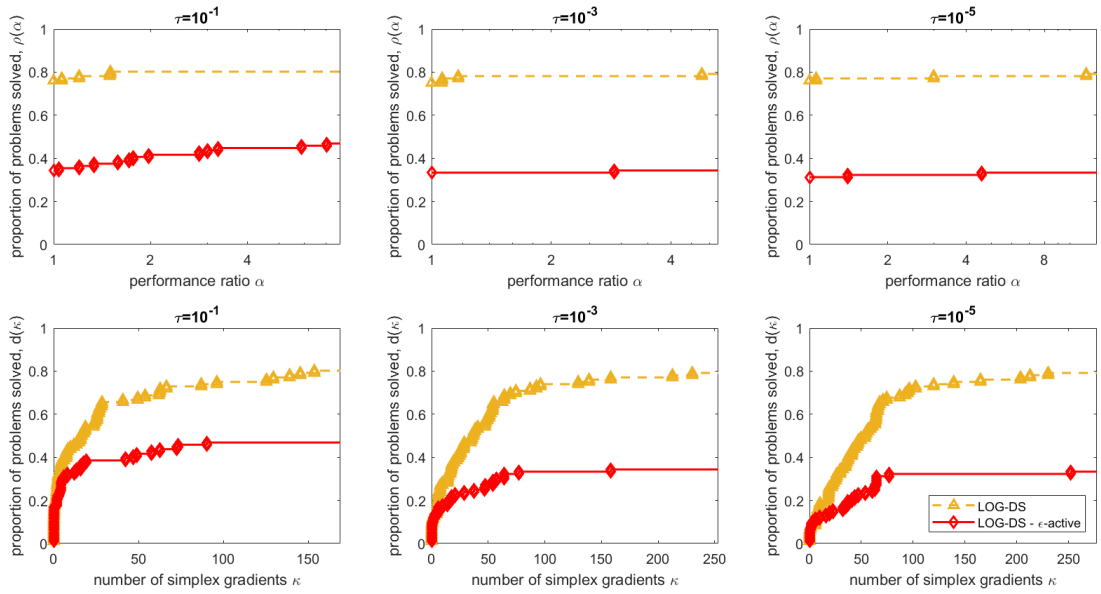


Figure 6.1: Performance (on top) and data (on bottom) profiles comparing LOG-DS using two different approaches to address linear inequality constraints.

6.1.3.2 Comparison with the original SID-PSM algorithm

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 is typically used for problems without equality constraints and for which a 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 6.1.

Figure 6.2 presents the comparison between LOG-DS, which exploits the mixed penalty-logarithmic barrier, 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 stepsize equal to 10^{-8}.

As Figure 6.2 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, regarding some of the constraints, allows to handle a wider class of practical problems.

6.1.3.3 Comparison with state-of-the-art solvers

This subsection focuses on comparing LOG-DS against state-of-the-art derivative-free optimization solvers that are able to address general nonlinear constraints. Comparisons were made with MADS [13], implemented in the well-known NOMAD package (version 4), which can be freely obtained at `https://www.gerad.ca/en/software/nomad`.

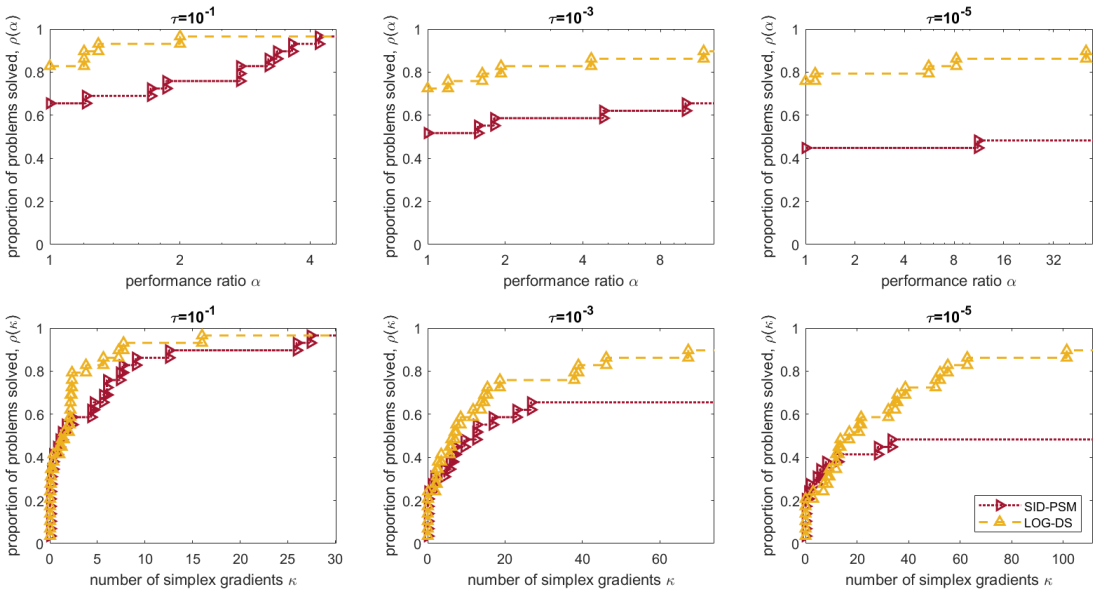


Figure 6.2: Performance (on top) and data (on bottom) profiles comparing LOG-DS and SID-PSM.

Additionally, the X-LOG-DFL algorithm [39], available through the DFL library as the LOGDFL package at `https://github.com/DerivativeFreeLibrary/LOGDFL`, was also tested. Comparison with X-LOG-DFL [39] is particularly relevant since it uses the same merit function of LOG-DS. Default settings were considered for all codes and results, reported in Figure 6.3, were obtained for a maximum 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 6.4 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.

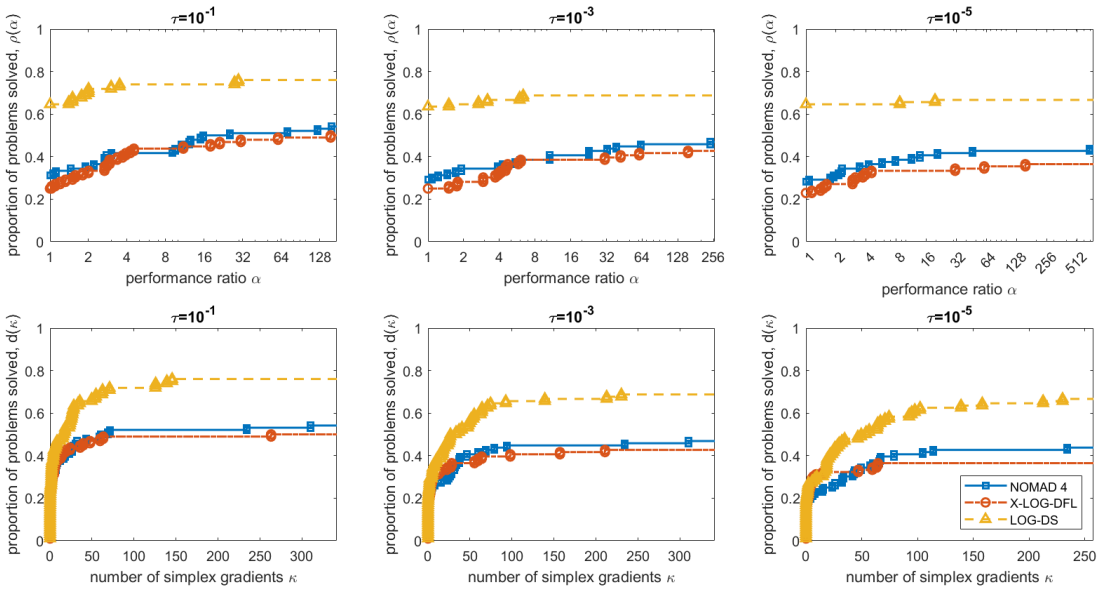


Figure 6.3: Performance (on top) and data (on bottom) comparing LOG-DS, NOMAD, and X-LOG-DFL, on the complete problem collection.

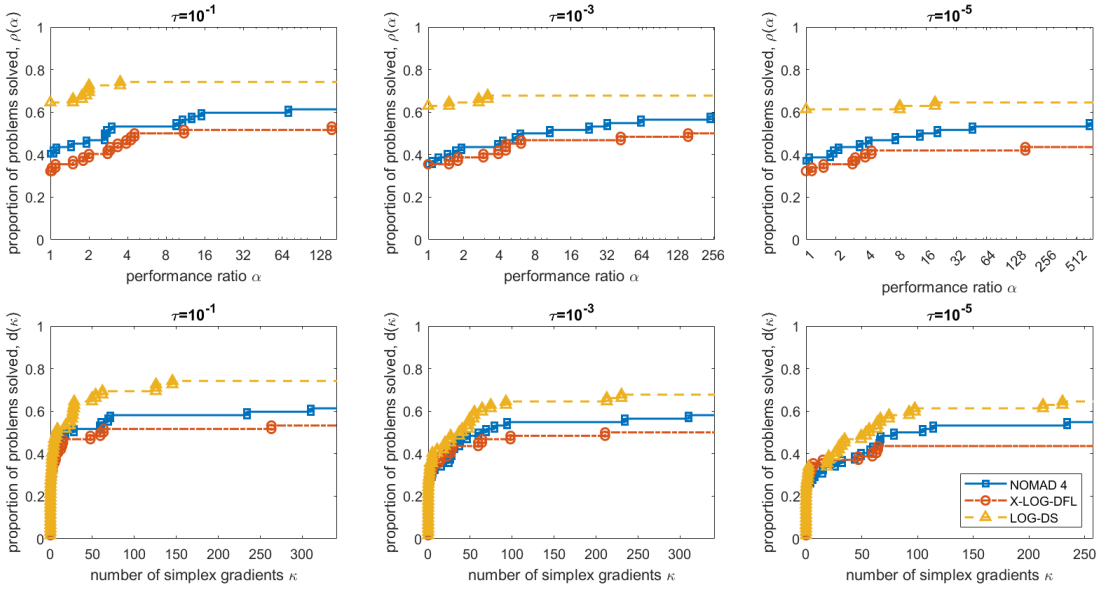


Figure 6.4: Performance (on top) and data (on bottom) profiles comparing LOG-DS, NOMAD, and X-LOG-DFL, on the subset of problems with only inequality constraints.

6.2 Extension to the multiobjective case

Considering the excellent results obtained with the logarithmic barrier approach for single-objective optimization, in this section, we propose its extension to inequality-constrained multiobjective derivative-free optimization, incorporated under the framework of Direct

Multisearch (DMS). This leads to the development of LOG-DMS, a direct search multiobjective optimization method that explicitly handles nonlinear inequality constraints using a logarithmic barrier approach.

To the best of our knowledge, [93] is the only work proposing a multiobjective logarithmic barrier method, which extends the scalar-valued internal penalty method to constrained multiobjective problems. This approach introduces a penalty barrier for the feasible set and relies on an auxiliary monotonic real-valued mapping that generates a sequence converging to a Pareto or weak Pareto optima, under mild assumptions. However, the proposed algorithm is not derivative-free, does not attempt to compute the complete Pareto front, and no numerical experiments have been conducted to demonstrate the method's effectiveness.

It is important to emphasize that the present section serves only to illustrate the viability of using the logarithmic barrier in a multiobjective derivative-free context. The primary goal here is to provide a proof-of-concept, rather than a comprehensive theoretical or empirical evaluation of the algorithm's properties. Consequently, while the initial results indicate that the approach can effectively handle constraints within multiobjective optimization, this section does not include formal proofs of the algorithm's convergence properties or other theoretical guarantees. Further research and more detailed analysis will be required to rigorously establish the full range of properties and performance of the proposed method.

Thus Section 6.2.1 defines the problem under analysis and details the proposed algorithmic structure. Section 6.2.2 provides information regarding the numerical implementation considered. Preliminary numerical experiments are reported in Section 6.2.3.

6.2.1 Problem definition and algorithmic structure

Consider the following multiobjective optimization problem with general inequality constraints, given by:

$$\begin{aligned} \min\ & \boldsymbol{F}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})\big)^\top \\ \text{s.t.}\ & \mathbf{g}(\boldsymbol{x}) \leq \mathbf{0} \\ & \boldsymbol{x} \in X \end{aligned} \tag{6.31}$$

where $\boldsymbol{F} : X \subseteq \mathbb{R}^n \to \{\mathbb{R} \cup \{+\infty\}\}^m$, $\mathbf{g} : X \subseteq \mathbb{R}^n \to \{\mathbb{R} \cup \{+\infty\}\}^p$, and $X := \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{lb} \leq \boldsymbol{x} \leq \mathbf{ub}\}$, with $\mathbf{lb}$, $\mathbf{ub} \in \mathbb{R}^n$ and $\mathbf{lb} < \mathbf{ub}$. We assume that the functions involved ($\boldsymbol{F}$ and $\mathbf{g}$) are Lipschitz continuous. Additionally, since X is a compact set, the feasible region $\mathcal{F}$, assumed to be nonempty and defined as

$$\mathcal{F} = X \cap \Omega \neq \emptyset,$$

with $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}\}$, is also a compact set.

In the strategy proposed in this section, the inequality constraints defining Ω are handled using a logarithmic barrier method, whereas the constraints defining X are addressed with an extreme barrier approach, ensuring that only points in X are evaluated.

Given a penalty parameter $\rho > 0$, we define, for each $\ell = 1, \ldots, m$, the merit function

$$Z_\ell(\boldsymbol{x}; \rho) = \begin{cases} f_\ell(\boldsymbol{x}) - \rho \sum_{i=1}^{p} \log\big(-g_i(\boldsymbol{x})\big) & \text{if } \boldsymbol{x} \in X \text{ and } \max_{i=1,\ldots,p} g_i(\boldsymbol{x}) < 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{6.32}$$

Let

$$\boldsymbol{Z}(\boldsymbol{x}; \rho) = \big(Z_1(\boldsymbol{x}; \rho),\ Z_2(\boldsymbol{x}; \rho),\ \ldots,\ Z_m(\boldsymbol{x}; \rho)\big)^\top$$

denote the corresponding vector function.

At each iteration k, given the barrier parameter $\rho_k > 0$, we consider the following problem:

$$\begin{aligned} &\min \ \ \boldsymbol{Z}(\boldsymbol{x}; \rho_k) = (Z_1(\boldsymbol{x}; \rho_k), Z_2(\boldsymbol{x}; \rho_k), \ldots, Z_m(\boldsymbol{x}; \rho_k))^\top \\ &\text{s.t. } \boldsymbol{x} \in X, \end{aligned} \tag{6.33}$$

where a suitable strategy is employed to update the parameter ρ_k. We also assume that a point $\boldsymbol{x}_0 \in X \cap \mathring{\Omega}$ is known, so that the set $X \cap \mathring{\Omega}$ is nonempty.

A common assumption to ensure the existence of an optimal solution for Problem (6.33) is the compactness of the lower-level sets of the objective function. In our approach, this translates to requiring that the sets

$$\mathcal{L}(\xi) := \bigcup_{\ell=1}^{m} \{\boldsymbol{x} \in X \mid Z_\ell(\boldsymbol{x}; \rho) \leq \xi\}$$

are compact for all $\rho > 0$. It is important to note that the function $\boldsymbol{Z}(\boldsymbol{x}; \rho)$ is only evaluated at points within X. To establish this property, we introduce the following assumption and subsequently prove the result in Lemma 6.2.2.

Assumption 6.2.1. *A point $\boldsymbol{x}_0 \in X$ exists such that $g_i(\boldsymbol{x}_0) < 0$ for all $i \in \{1, \ldots, p\}$.*

Lemma 6.2.2. *Let Assumption 6.2.1 hold. Then, for all $\rho > 0$ and $\xi \in \mathbb{R}$, the lower-level set*

$$\mathcal{L}(\xi) := \bigcup_{\ell=1}^{m} \{\boldsymbol{x} \in X \mid Z_\ell(\boldsymbol{x}; \rho) \leq \xi\}$$

is compact.

Proof. By definition, we have $\mathcal{L}(\xi) \subseteq X$. Thus, $\mathcal{L}(\xi)$ is bounded.

Now, consider any sequence $\{\boldsymbol{x}_k\} \in \mathcal{L}(\xi)$, such that $\lim_{k \to +\infty} \boldsymbol{x}_k = \bar{\boldsymbol{x}}$. By definition of $\mathcal{L}(\xi)$, for each $\boldsymbol{x}_k$, there is an index $\ell(k) \in \{1, \ldots, m\}$, such that

$$Z_{\ell(k)}(\boldsymbol{x}_k; \rho) \leq \xi,$$

Taking the limit for $k \to +\infty$ in the above relation, we get

$$\lim_{k \to +\infty} Z_{\ell(k)}(\boldsymbol{x}_k; \rho) \leq \xi.$$

Now, since the number of components of objective functions is finite, there exists at least one index ℓ^* that appears infinitely often in the sequence $\{\ell(k)\}$. This means that we can extract a subsequence $\{\boldsymbol{x}_{k_j}\}$ where $\ell(k_j) = \ell^*$ for all j. Taking limits along this subsequence, we obtain:

$$\lim_{j \to +\infty} Z_{\ell^*}(\boldsymbol{x}_{k_j}; \rho) = Z_{\ell^*}(\bar{\boldsymbol{x}}; \rho) \leq \xi. \tag{6.34}$$

Thus, $\bar{\boldsymbol{x}} \in \mathcal{L}(\xi)$, showing that the set is closed. This completes the proof. □

Algorithm 12 introduces a new variant of the DMS algorithm, designed to solve nonlinearly constrained, derivative-free multiobjective optimization problems, using the merit function defined in (6.32).

6.2.2 Implementation details

A first version of LOG-DMS was implemented in `Matlab`, preserving the default settings of DMS [64]. Specifically, the stepsize was initialized as 1 and was halved after unsuccessful iterations ($\beta_1 = \beta_2 = 0.5$). In successful iterations, unlike DMS, we double the stepsize ($\gamma = 2$). Regarding the logarithmic barrier parameter, we initialize it as $\rho_0 = 1$ and considered $\eta = 1$ and $\theta = 10^{-2}$ at Step 5 of the iterative process.

To promote a diverse spread of points in the Pareto front approximation generated by the solver, at Step 1 of Algorithm 12, an iterate point is selected based on the gamma spread (see equation (3.11)), as in the original DMS implementation. This point is then explored during the search step and, possibly, in the pool step.

After selecting the point at the beginning of iteration k, a search step may be performed in an attempt to find a new nondominated point with respect to $\boldsymbol{Z}(\cdot; \rho_k)$. Note that this step is optional, as the convergence properties of the method rely on the poll step (in our implementation, no search step was performed neither for LOG-DMS nor DMS). If the search step fails, the poll step is executed. A complete polling approach is adopted, evaluating all points corresponding to directions in the positive spanning set. An asymptotic dense set of directions in the unit sphere was considered, based on Sobol sequences [1], for both DMS and LOG-DMS.

In practical derivative-free optimization, the function evaluation is often the most computationally expensive step. To address this, DMS employs a simple caching strategy. Before computing a function value, it checks, using the infinity norm, if a nearby point, within a tolerance of 10^{-3}, has already been evaluated. If a nearby point is found, the evaluation is skipped. The same strategy is employed in LOG-DMS, which maintains a cache for the points, $\boldsymbol{P}^{\text{cache}} \in \mathbb{R}^{n \times |\text{cache}|}$, for the corresponding values of the objective function, $\boldsymbol{F}^{\text{cache}} \in \mathbb{R}^{m \times |\text{cache}|}$, and constraints, $\boldsymbol{G}^{\text{cache}} \in \mathbb{R}^{p \times |\text{cache}|}$.

If the penalty parameter is updated, we use the cache to recompute $\boldsymbol{Z}(\cdot; \rho_{k+1})$, taking into account the new penalty parameter as well as the values of $\boldsymbol{F}$ and $\boldsymbol{g}$ stored. Consequently, no new evaluations are needed during this process. After recomputing $\boldsymbol{Z}(\cdot; \rho_{k+1})$,

Algorithm 12 Logarithmic barrier in Direct Multisearch – LOG-DMS

Initialization

Choose $\boldsymbol{x}_0 \in X \cap \mathring{\Omega}$, $\alpha_0 > 0$ an initial stepsize parameter, $0 < \beta_1 \leq \beta_2 < 1$ the coefficients for stepsize contraction and $\gamma \geq 1$ the coefficient for stepsize expansion. Let $\mathcal{D}$ be a (possibly infinite) set of positive spanning sets, with directions $\boldsymbol{d}$ satisfying $0 < d_{\min} \leq \|\boldsymbol{d}\| \leq d_{\max}$, $\rho > 0$, $\theta \in (0,1)$, and $\eta \geq 1$. Initialize the list of nondominated points and corresponding stepsize parameters $L_0 = \{(\boldsymbol{x}_0; \alpha_0)\}$.

For $k = 0, 1, 2, \ldots$

1. **Selection of an iterate point:** Order the list L_k according to some criteria and select the first item $(\boldsymbol{x}; \alpha) \in L_k$ as the current iterate and stepsize parameter (thus setting $(\boldsymbol{x}_k; \alpha_k) = (\boldsymbol{x}; \alpha)$).

2. **Search Step (Optional):** Compute a finite set of points $\{\boldsymbol{z}_s\}_{s \in S}$ (in a mesh if $\bar{\rho}(\cdot) \equiv 0$) and evaluate $\mathbf{Z}$ in each element of $\{\boldsymbol{z}_s\}_{s \in S}$. Set $L_{\texttt{add}} = \{(\boldsymbol{z}_s; \alpha_k), s \in S\}$. Call $L_{\texttt{trial}} = \texttt{filter}(L_{\texttt{add}}, L_k)$ to remove all dominated points from $L_k \cup L_{\texttt{add}}$ (see Algorithm 6). If $L_{\texttt{trial}} \neq L_k$, declare the iteration as successful, set $L_{k+1} = L_{\texttt{trial}}$, and skip the poll step.

3. **Poll step:** Choose a positive spanning set D_k from the set $\mathcal{D}$. Evaluate $\mathbf{Z}$ at the set of poll points $P_k = \{\boldsymbol{x}_k + \alpha_k \boldsymbol{d} \mid \boldsymbol{d} \in D_k\}$. Set $L_{\texttt{add}} = \{(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k), \boldsymbol{d} \in D_k\}$. Call $L_{\texttt{trial}} = \texttt{filter}(L_{\texttt{add}}, L_k)$ to remove all dominated points from $L_k \cup L_{\texttt{add}}$ (see Algorithm 6). If $L_{\texttt{trial}} \neq L_k$, declare the iteration as successful and define $L_{k+1} = L_{\texttt{trial}}$. Otherwise, declare the iteration as unsuccessful and set $L_{k+1} = L_k$.

4. **Stepsize parameter update:** If the iteration was successful, then maintain or increase the corresponding stepsize parameter, by considering $\alpha_{k,\texttt{new}} \in [\alpha_k, \gamma \alpha_k]$. Replace all the new points $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}; \alpha_{k,\texttt{new}})$, when success is coming from the poll step, or $(\boldsymbol{z}_s; \alpha_k)$ in L_{k+1} by $(\boldsymbol{z}_s; \alpha_{k,\texttt{new}})$, when success is coming from the search step. Replace also $(\boldsymbol{x}_k; \alpha_k)$, if in L_{k+1}, by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$. Otherwise, decrease the stepsize parameter by choosing $\alpha_{k,\texttt{new}} \in [\beta_1 \alpha_k, \beta_2 \alpha_k]$, and replace the poll pair $(\boldsymbol{x}_k; \alpha_k)$ in L_{k+1} by $(\boldsymbol{x}_k; \alpha_{k,\texttt{new}})$.

5. **Penalty parameter update:** If the iteration was successful set $\rho_{k+1} = \rho_k$. Otherwise, define $(g_{\min})_k = \min\limits_{\boldsymbol{x} \in L_k} \left\{ \min\limits_{i=1,\ldots,p} \{|g_i(\boldsymbol{x})|\} \right\}$ and if $\alpha_{k+1} \leq \min\{\rho_k^{\eta}, (g_{\min})_k^2\}$, then decrease the penalty parameter, by setting $\rho_{k+1} = \theta \rho_k$.

EndFor

a new assessment of nondominated points is performed on the list. This step is required in order to maintain a list of nondominated points at each iteration, for function $\boldsymbol{Z}(\cdot; \rho_k)$.

6.2.3 Preliminary numerical experiments

In this section, we present numerical examples of applying LOG-DMS to five test problems and compare its performance with DMS (which employs an extreme barrier approach to address constraints) and DMS-FILTER-IR, the method proposed in Chapter 4, that explicitly handles constraints using a filter approach.

DMS and DMS-FILTER-IR were run using their default settings. For DMS, the best version identified in [64] was considered, with the exception of using Sobol sequences to generate a deterministic and asymptotic dense set of poll directions. Regarding DMS-FILTER-IR, it was run using the best version reported in Chapter 4. Results were obtained with a maximum budget of 5000 function evaluations.

Since DMS initialization requires a feasible point, while LOG-DMS initialization requires a strictly feasible point, these methods were initialized with the same strictly feasible point, to ensure a fair comparison. DMS-FILTER-IR was initialized as in Chapter 4, with n points evenly spaced in a line joining the lower and upper bounds on the variables. This initialization allows us to evaluate the performance of each method under identical conditions, providing a consistent basis for analyzing their respective convergence behaviors and solution quality.

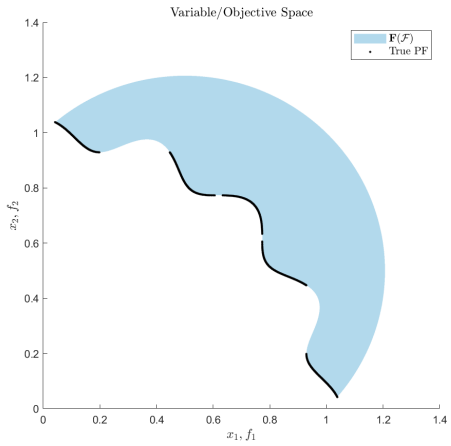
Example 6.2.3. *Consider the following problem, originally proposed in [178, Problem 2]:*

$$\begin{aligned} \min\ & F(\boldsymbol{x}) = (x_1, x_2)^\top \\ s.t.\ & g_1(\boldsymbol{x}) = -x_1^2 - x_2^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x_1}{x_2}\right)\right) \leq 0 \\ & g_2(\boldsymbol{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \\ & \boldsymbol{x} \in X = [0, \pi] \times [0, \pi] \end{aligned} \tag{TNK}$$

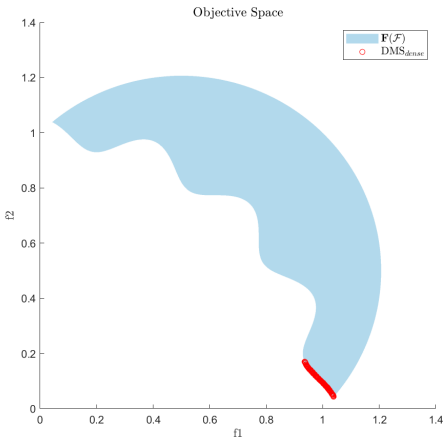
Since $f_1(x_1, x_2) = x_1$ and $f_2(x_1, x_2) = x_2$, the image set is identical to the feasible region. It is easy to see that every Pareto optimal solution lies on the boundary of the first constraint. However, because the first constraint function is periodic and the second constraint must also be satisfied, not every point on this boundary is Pareto optimal, leading to a disconnected Pareto front (see Figure 6.5(a)).

The initial point for Problem (TNK) was $\boldsymbol{x}_0 = (1.1, 0.15)^\top$. In Figure 6.5, we observe the results obtained by DMS (Figure 6.5(b)), LOG-DMS (Figure 6.5(c)), and DMS-FILTER-IR (Figure 6.5(d)). The LOG-DMS method stood out as superior approach, demonstrating greater efficiency in converging to Pareto optimal solutions and more comprehensively exploring the diversity of the Pareto set.

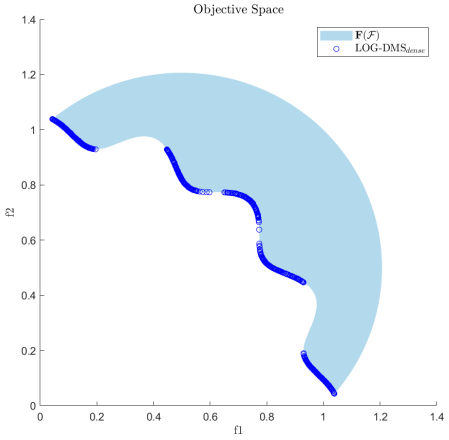
In [139], a set of 14 problems is proposed, extracted from the real world, that exhibits diverse characteristics. Among these 14 problems, we selected 4 biobjective problems with a connected feasible region. To facilitate notation, consider the two distance functions –



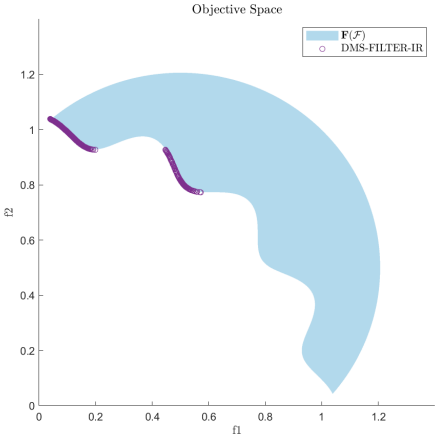
(a) True Pareto front of Problem (TNK).



(b) Approximation generated by DMS.



(c) Approximation generated by LOG-DMS.



(d) Approximation generated by DMS-FILTER-IR.

Figure 6.5: Approximations to the Pareto front generated by DMS, LOG-DMS, and DMS-FILTER-IR for Problem (TNK).

the biased distance function $d_1(\boldsymbol{x})$ and the distance function with variable linkages $d_2(\boldsymbol{x})$ – defined as:

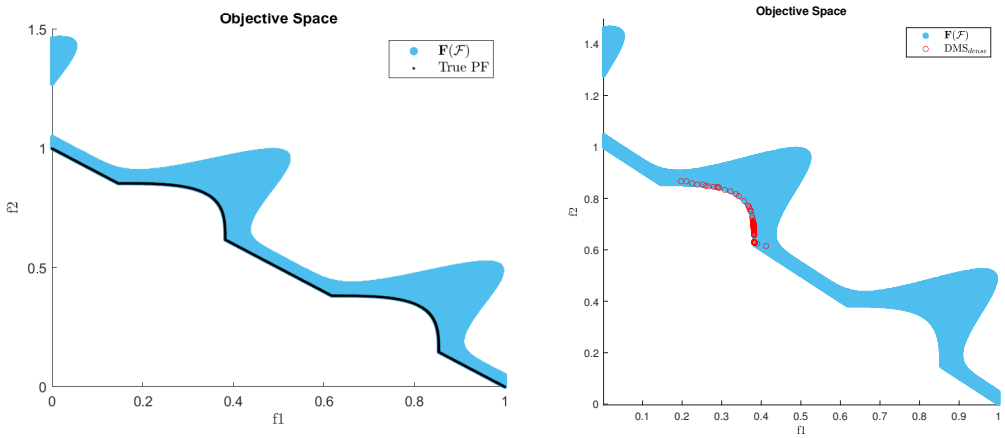
$$
\begin{aligned}
d_1(\boldsymbol{x}) &= 1 + \sum_{i=2}^{n} \left(1 - \exp\left(-10 \left(x_i^{n-2} - 0.5 - \frac{i-1}{2n} \right)^2 \right) \right), \\
d_2(\boldsymbol{x}) &= 1 + \sum_{i=2}^{n} 2 \left(x_i + (x_{i-1} - 0.5)^2 - 1 \right)^2 .
\end{aligned}
$$

In the following problems we set $n = 15$.

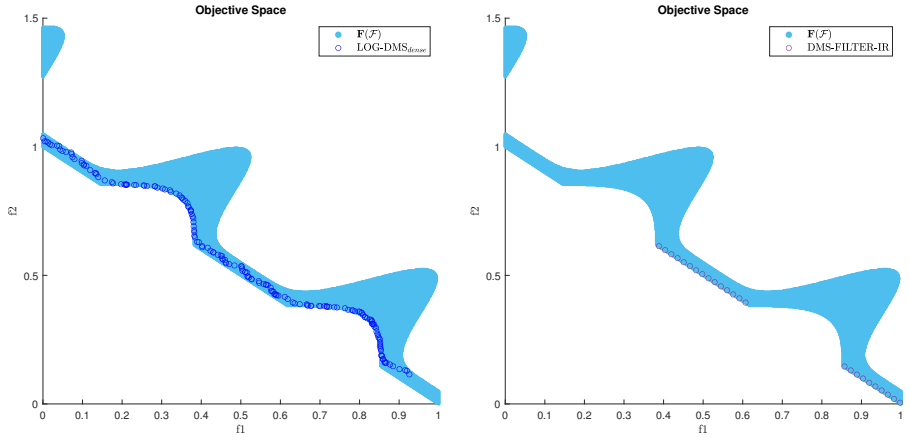
Example 6.2.4. *Consider the following problem, originally proposed in [139, Problem MW3]:*

$$\begin{aligned} \min\ & F(\boldsymbol{x}) = (x_1, d_2(\boldsymbol{x}) - x_1)^\top \\ s.t.\ & g_1(\boldsymbol{x}) = d_2(\boldsymbol{x}) - 0.45\sin(0.75\pi\phi(\boldsymbol{x}))^6 - 1.05 \leq 0, \\ & g_2(\boldsymbol{x}) = 0.85 - d_2(\boldsymbol{x}) + 0.3\sin(0.75\pi\phi(\boldsymbol{x}))^2 \leq 0, \\ & \phi(\boldsymbol{x}) = \sqrt{2}\,(d_1(\boldsymbol{x}) - 2x_1)\,, \\ & \boldsymbol{x} \in [0,1]^n. \end{aligned} \tag{MW3}$$

In Figure 6.6(a), the blue area represents the image of the feasible region in the objective space, and the black curve denotes the Pareto front. Since Problem (MW3) includes some narrow parts in its feasible region, finding the Pareto optimal solutions in these areas is particularly challenging.



(a) True Pareto front of Problem (MW3). (b) Approximation generated by DMS.



(c) Approximation generated by LOG-DMS. (d) Approximation generated by DMS-FILTER-IR.

Figure 6.6: Approximations to the Pareto front generated by DMS, LOG-DMS, and DMS-FILTER-IR for Problem (MW3).

Example 6.2.5. *Consider the following problem, originally proposed in [139, Problem MW5]:*

$$\begin{aligned}
\min\ & F(\boldsymbol{x}) = \left(d_1(\boldsymbol{x})x_1, d_1(\boldsymbol{x})\sqrt{1-x_1^2}\right)^\top \\
s.t.\ & g_1(\boldsymbol{x}) = d_1(\boldsymbol{x})^2 - \left(1.7 - 0.2\sin(2\phi_1(\boldsymbol{x}))\right)^2 \leq 0, \\
& g_2(\boldsymbol{x}) = \left(1 + 0.5\sin(6\phi_2(\boldsymbol{x})^2)\right)^2 - d_1(\boldsymbol{x})^2 \leq 0, \\
& g_3(\boldsymbol{x}) = \left(1 - 0.45\sin(6\phi_2(\boldsymbol{x})^3)\right)^2 - d_1(\boldsymbol{x})^2 \leq 0, \\
& \phi_1(\boldsymbol{x}) = \arctan\left(\frac{\sqrt{1-x_1^2}}{x_1}\right), \\
& \phi_2(\boldsymbol{x}) = 0.5\pi - 2\left|\arctan\left(\frac{\sqrt{1-x_1^2}}{x_1}\right) - 0.25\pi\right|, \\
& \boldsymbol{x} \in [0,1]^n.
\end{aligned} \tag{MW5}$$

The difficulty of Problem (MW5) lies in the fact that the images of the discrete Pareto optimal solutions are located at the end of the tunnel-like region, as we can see in Figure 6.7(a).

Example 6.2.6. *Consider the following problem, originally proposed in [139, Problem MW7]:*

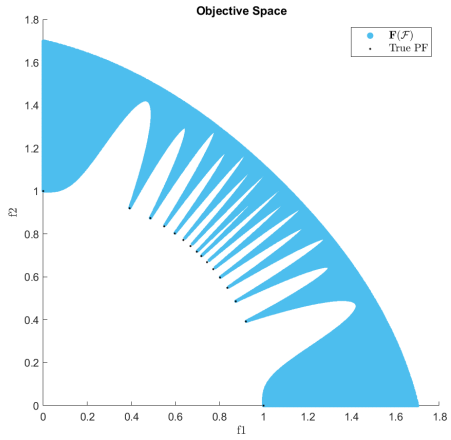
$$\begin{aligned}
\min\ & F(\boldsymbol{x}) = \left(d_2(\boldsymbol{x})x_1, d_2(\boldsymbol{x})\sqrt{1-x_1^2}\right)^\top \\
s.t.\ & g_1(\boldsymbol{x}) = d_2(\boldsymbol{x})^2 - \left(1.2 + 0.4\sin(4\phi_1(\boldsymbol{x}))^{16}\right)^2 \leq 0, \\
& g_2(\boldsymbol{x}) = \left(1.15 - 0.2\sin(4\phi(\boldsymbol{x}))^8\right)^2 - d_2(\boldsymbol{x})^2 \leq 0, \\
& \phi(\boldsymbol{x}) = \arctan\left(\frac{\sqrt{1-x_1^2}}{x_1}\right), \\
& \boldsymbol{x} \in [0,1]^n.
\end{aligned} \tag{MW7}$$

Figure 6.8(a) illustrates the image of the feasible region for Problem (MW7). This problem presents several challenges due to its highly nonlinear constrained structure and the complex shape of its feasible region. In particular, the Pareto front is located in difficult-to-reach areas, and the nonlinear constraints yield a tunnel-like shape that further complicates the discovery of Pareto optimal solutions.

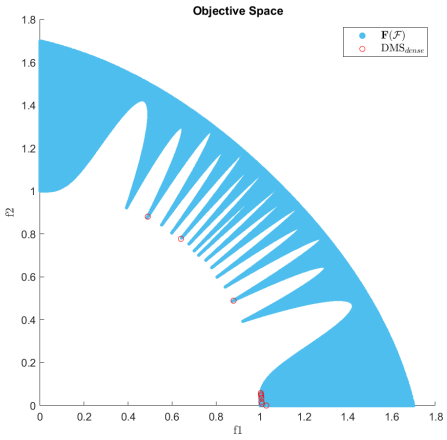
Example 6.2.7. *Consider the following problem, originally proposed in [139, Problem MW9]:*

$$\begin{aligned}
\min\ & F(\boldsymbol{x}) = \left(d_1(\boldsymbol{x})x_1, d_1(\boldsymbol{x})\left(1 - x_1^{0.6}\right)\right)^\top \\
s.t.\ & g_1(\boldsymbol{x}) = \min\{T_1(\boldsymbol{x}), T_2(\boldsymbol{x}), T_3(\boldsymbol{x})\} \leq 0, \\
& T_1(\boldsymbol{x}) = \left(1 - 0.64d_1(\boldsymbol{x})^2x_1^2 - d_1(\boldsymbol{x})\left(1 - x_1^{0.6}\right)\right)\left(1 - 0.36d_1(\boldsymbol{x})^2x_1^2 - d_1(\boldsymbol{x})\left(1 - x_1^{0.6}\right)\right), \\
& T_2(\boldsymbol{x}) = 1.35^2 - \left(d_1(\boldsymbol{x})x_1 + 0.35\right)^2 - d_1(\boldsymbol{x})\left(1 - x_1^{0.6}\right), \\
& T_3(\boldsymbol{x}) = 1.15^2 - \left(d_1(\boldsymbol{x})x_1 + 0.15\right)^2 - d_1(\boldsymbol{x})\left(1 - x_1^{0.6}\right), \\
& \boldsymbol{x} \in [0,1]^n.
\end{aligned} \tag{MW9}$$

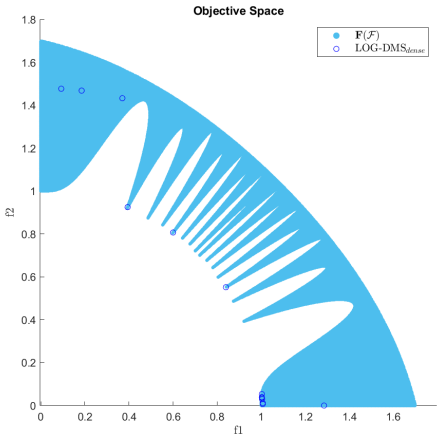
Figure 6.9(a) illustrates the image of the feasible region in the objective space along with the true Pareto front. The narrow, tunnel-like structure of the region, which results from the problem's nonconvexity, nonlinearity, and nonsmooth constraints, highlights the inherent challenges of this optimization task. Notably, the Pareto-optimal solutions are



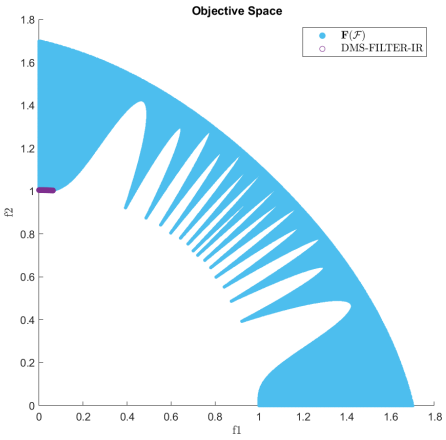
(a) True Pareto front of Problem (MW5).



(b) Approximation generated by DMS.



(c) Approximation generated by LOG-DMS.



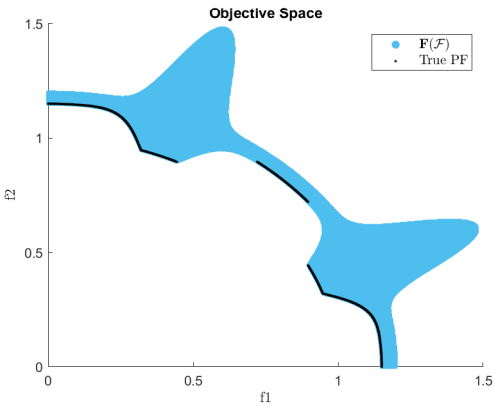
(d) Approximation generated by DMS-FILTER-IR.

Figure 6.7: Approximations to the Pareto front generated by DMS, LOG-DMS, and DMS-FILTER-IR for Problem (MW5).

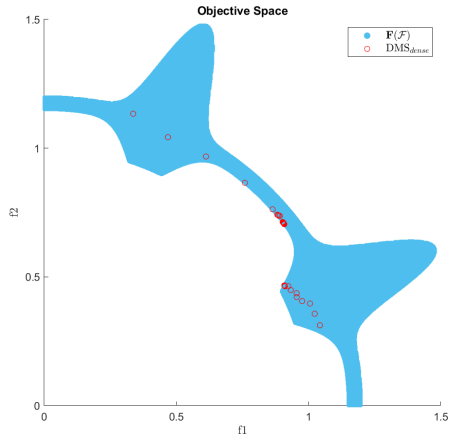
confined to the tunnel's terminal end, reflecting the difficulty of identifying high-quality trade-offs.

Table 6.2 presents the hypervolume (see equation (3.10)) values obtained for the five different test problems using three distinct optimization approaches: DMS, LOG-DMS, and DMS-FILTER-IR. The corresponding approximations to the Pareto fronts can be found in Figures 6.5-6.9.

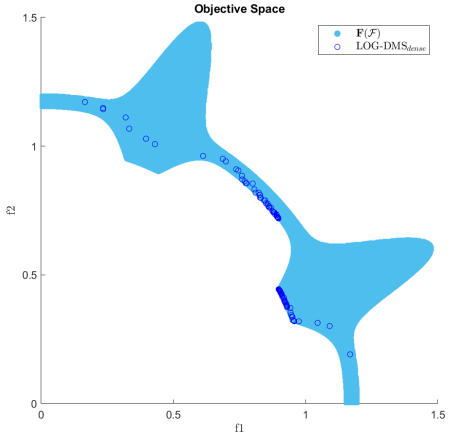
In conclusion, the numerical experiments conducted on five challenging test problems demonstrate the viability of incorporating a logarithmic barrier approach within the direct multisearch framework for addressing multiobjective constrained optimization problems. LOG-DMS generally outperforms the traditional DMS and the filter-based DMS-FILTER-IR in terms of hypervolume, indicating a promising exploration of the Pareto front under inequality constraints. While these initial results serve as proof-of-concept for



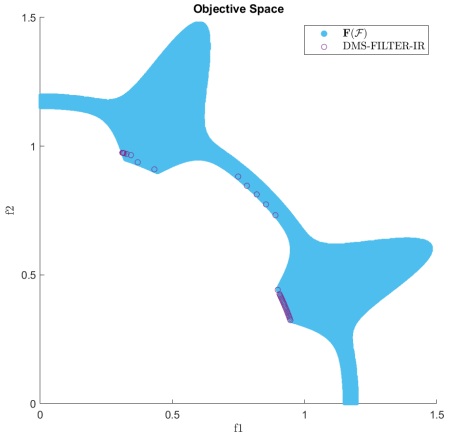
(a) True Pareto front of Problem (MW7).



(b) Approximation generated by DMS.



(c) Approximation generated by LOG-DMS.



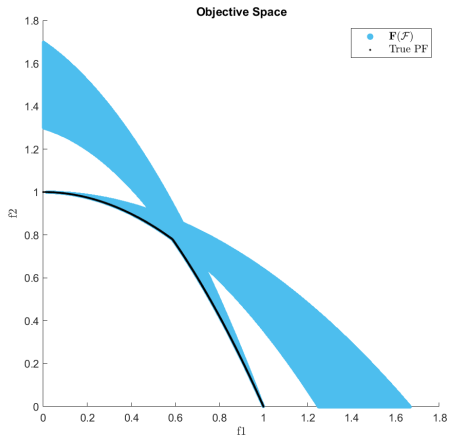
(d) Approximation generated by DMS-FILTER-IR.

Figure 6.8: Approximations to the Pareto front generated by DMS, LOG-DMS, and DMS-FILTER-IR for Problem (MW7).

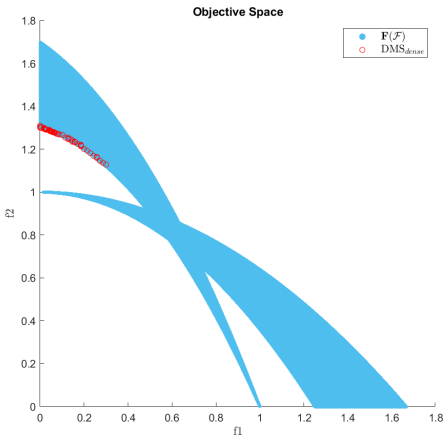
Problem	DMS	LOG-DMS	DMS-FILTER-IR
TNK	9.6%	30.81%	18.97%
MW3	28.70%	45.18%	39.71%
MW5	41.91%	44.41%	32.11%
MW7	31.93%	36.20%	39.41%
MW9	32.85%	60.07%	44.79%

Table 6.2: Hypervolume values obtained for the five test problems using various optimization approaches.

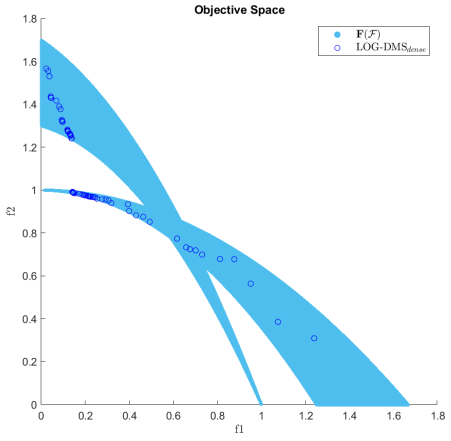
handling nonlinear constraints via a logarithmic barrier, they also highlight the need for further comprehensive studies to rigorously assess convergence properties and overall performance in broader settings.



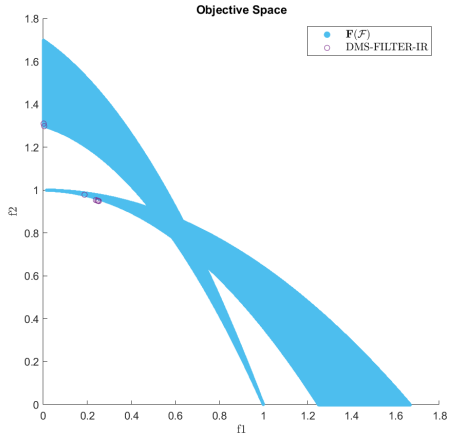
(a) True Pareto front of Problem MW9.



(b) Approximation generated by DMS.



(c) Approximation generated by LOG-DMS.



(d) Approximation generated by DMS-FILTER-IR.

Figure 6.9: Approximations to the Pareto front generated by DMS, LOG-DMS, and DMS-FILTER-IR for Problem (MW9).

7

Conclusions and Future Work

The main goal of the present work was to develop efficient and robust algorithms suited for constrained multiobjective derivative-free optimization.

In Chapter 4, we proposed and analyzed a novel direct multisearch filter approach, combined with an inexact restoration method, denoted by DMS-FILTER-IR.This algorithm is designed to tackle constrained multiobjective derivative-free optimization problems without requiring a feasible initialization for the relaxable constraints. By employing a filter-based strategy, it overcomes the challenges of applying directional direct search when a feasible initialization is unavailable. Moreover, incorporating an inexact restoration step significantly enhanced the attainment of feasibility.

Under standard assumptions in derivative-free optimization analysis, we established the existence of at least one subsequence of iterates generated by the algorithm that converges to a Pareto-Clarke critical point of a related problem. Additional conditions were provided to guarantee the existence of a feasible point and ensure convergence to a Pareto-Clarke critical point for the original problem.

Extensive numerical experience allowed to establish the competitiveness of the proposed algorithm, by comparison with the original DMS and state-of-art solvers for constrained multiobjective derivative-free optimization. Filter methods combined with inexact restoration techniques offer a valuable alternative to penalty function methods or the use of a progressive barrier strategy, enhancing the practical performance of optimization algorithms, particularly in scenarios involving complex constraints and biobjective problems.

We have observed that a filter approach reformulates the original problem by aggregating constraint violations into an additional objective function component to be minimized, thereby increasing the number of components to be minimized by one. As a result of this filter-based strategy, DMS-FILTER-IR transforms a biobjective problem into one with a third objective, leveraging the well-established strength of DMS in effectively managing problems with either two or three objectives.

In Chapter 5, we introduced DMS-Reduction, a new variant that builds on a framework employing search directions selected from variable subspaces or derived from correlation

matrices that assess the relationships between variables and objectives. This new version of DMS is designed to tackle many-objective optimization problems. The reduction strategy addresses large-scale challenges by simultaneously decreasing the number of objective function components and the number of problem variables involved. Extensive numerical experiments demonstrate the competitiveness of this approach, demonstrates performance offering a promising pathway for addressing complex optimization challenges. In particular, DMS-Reduction could be an excellent departure point to incorporate the filter approaches developed in Chapter 4.

In Chapter 6, the primary objective was to extend the approach introduced in [39] to single-objective direct search, enabling the efficient handling of general constraints and acting as a first step to generalize it to a multiobjective setting.

Building on the SID-PSM algorithm, a generalized pattern search method that utilizes polynomial models in both the search and poll steps to enhance the numerical performance, we proposed a new algorithm, LOG-DS. While preserving the core algorithmic structure of SID-PSM, LOG-DS introduces a mixed penalty-logarithmic barrier merit function to handle general constraints. This function effectively separates inequality constraints, applying exterior penalties to some while employing a logarithmic barrier for others. This distinction enables LOG-DS to dynamically manage constraints throughout the optimization process, improving its ability to navigate feasible regions within complex landscapes.

Under standard assumptions, without requiring convexity of the functions defining the problem, the convergence of LOG-DS was established towards stationary points. This theoretical foundation highlights the algorithm's adaptability to a wide range of constrained problems, making it a powerful tool in constrained black-box optimization. Additionally, surrogate models enhance LOG-DS's performance by providing efficient merit function approximations, reducing unnecessary evaluations, and optimizing computational load, thereby accelerating convergence.

Furthermore, extensive numerical experiments were conducted to compare the performance of LOG-DS against several state-of-the-art solvers, including SID-PSM, on a large set of test problems. The numerical results demonstrate the robustness and efficiency of the proposed algorithm.

The initial analysis of LOG-DS provided a crucial assessment of the effectiveness of the logarithmic barrier approach, serving as a foundational step before extending it to the multiobjective setting. This initial validation confirmed that the barrier method shows promise, thereby justifying its adaptation for constrained multiobjective derivative-free optimization.

Therefore, we have proposed LOG-DMS, an extension of logarithmic barrier approach within a direct multisearch framework for constrained multiobjective derivative-free optimization. The numerical experiments conducted on five test problems demonstrate that LOG-DMS generally outperforms the traditional DMS and the filter-based DMS-FILTER-IR in terms of hypervolume.

Future research directions

Numerous promising directions for future research derive from the current work, spanning both algorithm development and theoretical analysis.

Several extensions can be considered to improve the numerical performance of DMS-FILTER-IR. For instance, one could define a search step based on surrogate models [37] or employ parallelism [179]. Since DMS-FILTER-IR served as the primary motivation for DMS-Reduction, a logical next step would be to implement both the inexact restoration phase and the filter approach into DMS-Reduction, to assess their performance in more than two objectives. Moreover, in a recent paper [43], the authors focus on constructing models in random, low-dimensional subspaces and achieve theoretical improvements by leveraging random embeddings. Given that the first version of DMS-Reduction does not include a search step, it would be interesting to incorporate such submodels into a search step to enhance its performance.

A second approach to constraints was considered in the present work, with the algorithm LOG-DS, now in single-objective constrained setting. Under smooth assumptions, we established the theoretical convergence properties of LOG-DS. However, direct search methods is know for their ability to handle nonsmooth functions. Therefore, it would be important to extend this analysis to cases where the smoothness assumption is relaxed.

Moreover, the good performance of LOG-DMS indicates a more effective exploration of the Pareto front under inequality constraints. While these results provide a promising proof-of-concept for handling nonlinear constraints using a logarithmic barrier, they also highlight the necessity for further comprehensive studies to rigorously assess convergence properties and overall performance in broader settings.

Bibliography

[1] M. A. Abramson, C. Audet, J. E. Dennis Jr., and S. Le Digabel. "OrthoMADS: A deterministic MADS instance with orthogonal directions". In: *SIAM J. Optim.* 20 (2009), pp. 948–966 (cit. on pp. 59, 64, 109).

[2] M. A. Abramson, O. A. Brezhneva, J. E. Dennis Jr., and R. L. Pingel. "Pattern search in the presence of degenerate linear constraints". In: *Optim. Methods Softw.* 23 (2008), pp. 297–319 (cit. on pp. 33, 91, 103).

[3] H. Afshari, W. Hare, and S. Tesfamariam. "Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures". In: *Appl. Soft Comp.* 83 (2019), p. 105631 (cit. on p. 40).

[4] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. "On augmented Lagrangian methods with general lower-level constraints". In: *SIAM J. Optim.* 18 (2007), pp. 1286–1309 (cit. on p. 32).

[5] R. Andreani, A. L. Custódio, and M. Raydan. "Using first-order information in Direct Multisearch for multiobjective optimization". In: *Optim. Methods Softw.* 37 (2022), pp. 2135–2156 (cit. on p. 44).

[6] M. A. Ansary and G. Panda. "A globally convergent SQCQP method for multiobjective optimization problems". In: *SIAM J. Optim.* 31 (2021), pp. 91–113 (cit. on p. 43).

[7] M. A. Ansary and G. Panda. "A modified quasi-Newton method for vector optimization problem". In: *Optimization* 64 (2015), pp. 2289–2306 (cit. on p. 40).

[8] P. B. Assunção, O. P. Ferreira, and L. F. Prudente. "Conditional gradient method for multiobjective optimization". In: *Comput. Optim. Appl.* 78 (2021), pp. 741–768 (cit. on p. 40).

[9] C. Audet, V. Béchard, and S. Le Digabel. "Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search". In: *J. Global Optim.* 41 (2008), pp. 299–318 (cit. on p. 82).

[10] C. Audet and J. E. Dennis Jr. "A pattern search filter method for nonlinear programming without derivatives". In: *SIAM J. Optim.* 14 (2004), pp. 980–1010 (cit. on pp. 34, 47, 51, 52, 55).

[11] C. Audet and J. E. Dennis Jr. "A progressive barrier for derivative-free nonlinear programming". In: *SIAM J. Optim.* 20 (2009), pp. 445–472 (cit. on pp. 34, 47).

[12] C. Audet and J. E. Dennis Jr. "Analysis of generalized pattern searches". In: *SIAM J. Optim.* 13 (2002), pp. 889–903 (cit. on pp. 37, 88).

[13] C. Audet and J. E. Dennis Jr. "Mesh adaptive direct search algorithms for constrained optimization". In: *SIAM J. Optim.* 17 (2006), pp. 188–217 (cit. on pp. 15, 16, 33–35, 47, 59, 104).

[14] C. Audet and W. Hare. *Derivative-free and Blackbox Optimization*. Cham, Switzerland: Springer, 2017 (cit. on pp. 2, 3, 13, 33, 37, 43, 59).

[15] C. Audet, S. Le Digabel, and M. Peyrega. "Linear equalities in blackbox optimization". In: *Comput. Optim. Appl.* 61 (2015), pp. 1–23 (cit. on p. 34).

[16] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. *The NOMAD project*. URL: https://www.gerad.ca/nomad (cit. on p. 34).

[17] C. Audet, G. Savard, and W. Zghal. "A mesh adaptive direct search algorithm for multiobjective optimization". In: *European J. Oper. Res.* 204 (2010), pp. 545–556 (cit. on pp. 44, 46, 82).

[18] C. Audet, G. Savard, and W. Zghal. "Multiobjective optimization through a series of single-objective formulations". In: *SIAM J. Optim.* 19 (2008), pp. 188–210 (cit. on pp. 24, 44, 46, 49).

[19] M. Avriel. *Nonlinear Programming: Analysis and Methods*. Originally published: Englewood Cliffs, N.J.: Prentice-Hall, 1976. Mineola, NY: Dover Publications, 2003 (cit. on p. 11).

[20] A. Bagirov, N. Karmitsa, and M. M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer Cham, 2014 (cit. on pp. 13–18).

[21] S. Bandyopadhyay, S. K. Pal, and B. Aruna. "Multiobjective GAs, quantitative indices, and pattern classification". In: *IEEE Trans. Syst. Man Cybern. - Part B: Cybern.* 34 (2004), pp. 2088–2099 (cit. on p. 48).

[22] A. Banerjee, S. Pradhan, B. Misra, and S. Chakraborty. "A guide to meta-heuristic algorithms for multi-objective optimization: concepts and approaches". In: *Applied Multi-objective Optimization*. Ed. by Nilanjan Dey. Singapore: Springer Nature Singapore, 2024, pp. 1–19 (cit. on pp. 2, 40).

[23] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2nd ed. CMS Books in Mathematics. Springer, Cham, 2017 (cit. on p. 8).

[24] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. 3rd ed. Wiley Publishing, 2006 (cit. on pp. 9, 11, 12).

[25] J. Y. Bello-Cruz, L. R. L. Pérez, and J. G. Melo. "Convergence of the projected gradient method for quasiconvex multiobjective optimization". In: *Nonlinear Anal.* 74 (2011), pp. 5268–5273 (cit. on p. 40).

[26] A. Benchakroun, J. Dussault, and A. Mansouri. "A two parameter mixed interior-exterior penalty algorithm". In: *ZOR - Meth. and Models of Oper. Res.* 41 (1995), pp. 25–55 (cit. on p. 31).

[27] D. P. Bertsekas. *Nonlinear Programming*. 2nd ed. Belmont, Massachusetts: Athena Scientific, 1999 (cit. on pp. 8, 13, 92).

[28] J. Bigeon, S. Le Digabel, and L. Salomon. "DMulti-MADS: Mesh adaptive direct multisearch for bound-constrained blackbox multiobjective optimization". In: *Comput. Optim. Appl.* 79 (2021), pp. 301–338 (cit. on pp. 46, 47, 50, 57).

[29] J. Bigeon, S. Le Digabel, and L. Salomon. "Handling of constraints in multiobjective blackbox optimization". In: *Comput. Optim. Appl.* 89 (2024), pp. 69–113 (cit. on pp. 46, 50, 53, 60, 64, 66, 84).

[30] E. G. Birgin and J. M. Martínez. "Large-scale active-set box-constrained optimization method with spectral projected gradients". In: *Comput. Optim. Appl.* 23 (2002), pp. 101–125 (cit. on p. 32).

[31] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Fundamentals of Algorithms. Society for Industrial & Applied Mathematics, 2014 (cit. on p. 32).

[32] E. G. Birgin, J. M. Martínez, and M. Raydan. "Inexact spectral projected gradient methods on convex sets". In: *IMA J. Numer. Anal.* 23 (2003), pp. 539–559 (cit. on p. 30).

[33] E. G. Birgin, J. M. Martínez, and M. Raydan. "Nonmonotone spectral projected gradient methods on convex sets". In: *SIAM J. Optim.* 10 (2000), pp. 1196–1211 (cit. on p. 30).

[34] 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*. Ed. by J. T. Borggaard, J. Burns, E. Cliff, and S. Schreck. Vol. 24. Birkhäuser, Boston, 1998, pp. 77–90 (cit. on pp. 38, 100).

[35] J. P. Boyle and R. L. Dykstra. "A method for finding projections onto the intersection of convex sets in Hilbert spaces". In: *Advances in Order Restricted Statistical Inference*. Ed. by R. L. Dykstra, T. Robertson, and F. T. Wright. New York, NY: Springer New York, 1986, pp. 28–47 (cit. on p. 30).

[36] J. Branke, K. Deb, K. Miettinen, and R. Slowiński, eds. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer-Verlag, 2008 (cit. on p. 49).

[37] C. P. Brás and A. L. Custódio. "On the use of polynomial models in multiobjective directional direct search". In: *Comput. Optim. Appl.* 77 (2020), pp. 897–918 (cit. on pp. 44, 53, 83, 120).

[38] 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. arXiv: 2407.21634 [math.OC] (cit. on pp. 5, 86).

[39] A. Brilli, G. Liuzzi, and S. Lucidi. "An interior point method for nonlinear constrained derivative-free optimization". In: *Optim. Methods Softw.* (2025), pp. 1–39. DOI: 10.1080/10556788.2025.2453110 (cit. on pp. 5, 34, 87, 90, 101, 105, 119).

[40] R. P. Brito, H. Sebastião, and P. Godinho. "Portfolio management with higher moments: the cardinality impact". In: *Int. Trans. in Oper. Res.* 26 (2019), pp. 2531–2560 (cit. on p. 40).

[41] P. H. Calamai and J. J. Moré. "Projected gradient methods for linearly constrained problems". In: *Math. Program.* 39 (1987), pp. 93–116 (cit. on p. 30).

[42] G. A. Carrizo, P. A. Lotito, and M. C. Maciel. "Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem". In: *Math. Program.* 159 (2016), pp. 339–369 (cit. on pp. 42, 43).

[43] C. Cartis and L. Roberts. "Scalable subspace methods for derivative-free nonlinear least-squares optimization". In: *Math. Program.* 199 (2023), pp. 461–524 (cit. on pp. 74, 120).

[44] W. Chen, X. Yang, and Y. Zhao. "Conditional gradient method for vector optimization". In: *Comput. Optim. Appl.* 85 (2023), pp. 857–896 (cit. on p. 40).

[45] T. D. Chuong. "Newton-like methods for efficient solutions in vector optimization". In: *Comput. Optim. Appl.* 54 (2013), pp. 495–516 (cit. on p. 40).

[46] F. H. Clarke. "A new approach to Lagrange multipliers". In: *Math. Oper. Res.* 1 (1976), pp. 165–174 (cit. on p. 15).

[47] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Philadelphia, USA: SIAM, 1990 (cit. on pp. 13–15).

[48] G. Cocchi and M. Lapucci. "An augmented Lagrangian algorithm for multi-objective optimization". In: *Comput. Optim. Appl.* 77 (2020), pp. 29–56 (cit. on p. 43).

[49] G. Cocchi, M. Lapucci, and P. Mansueto. "Pareto front approximation through a multi-objective augmented Lagrangian method". In: *EURO J. Comput. Optim.* 9 (2021), p. 100008 (cit. on p. 43).

[50] G. Cocchi, G. Liuzzi, S. Lucidi, and M. Sciandrone. "On the convergence of steepest descent methods for multiobjective optimization". In: *Comput. Optim. Appl.* 77 (2020), pp. 1–27 (cit. on p. 40).

[51] G. Cocchi, G. Liuzzi, A. Papini, and M. Sciandrone. "An implicit filtering algorithm for derivative-free multiobjective optimization with box constraints". In: *Comput. Optim. Appl.* 69 (2018), pp. 267–296 (cit. on pp. 47, 49).

[52] A. R. Conn. "Constrained optimization using a nondifferentiable penalty function". In: *SIAM J. Numer. Anal.* 10 (1973), pp. 760–784 (cit. on p. 31).

[53] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. "Global convergence of a class of trust region algorithms for optimization with simple bounds". In: *SIAM J. Numer. Anal.* 25 (1988), pp. 433–460 (cit. on p. 29).

[54] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust Region Methods*. Philadelphia: Society for Industrial and Applied Mathematics, 2000 (cit. on p. 38).

[55] A. R. Conn, K. Scheinberg, and Ph. L. Toint. "On the convergence of derivative-free methods for unconstrained optimization". In: *Approximation Theory and Optimization: Tributes to M. J. D. Powell*. Ed. by A. Iserles and M. Buhmann. Publication editors: A. Iserles and M. Buhmann. United Kingdom: Cambridge University Press, 1997, pp. 83–108 (cit. on p. 38).

[56] A. R. Conn, K. Scheinberg, and Ph. L. Toint. "Recent progress in unconstrained nonlinear optimization without derivatives". In: *Math. Program.* 79 (1997), pp. 397–414 (cit. on p. 38).

[57] A. R. Conn, K. Scheinberg, and L. N. Vicente. "Geometry of interpolation sets in derivative-free optimization". In: *Math. Program.* 111 (2008), pp. 141–172 (cit. on p. 38).

[58] A. R. Conn, K. Scheinberg, and L. N. Vicente. "Geometry of sample sets in derivative-free optimization: Polynomial regression and underdetermined interpolation". In: *IMA J. of Numer. Anal.* 28 (2008), pp. 721–748 (cit. on p. 38).

[59] A. R. Conn, K. Scheinberg, and L. N. Vicente. "Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points". In: *SIAM J. Optim.* 20 (2009), pp. 387–415 (cit. on p. 38).

[60] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-free Optimization*. MPS-SIAM Series on Optimization. Philadelphia, USA: SIAM, 2009 (cit. on pp. 2, 3, 13, 33, 37, 38, 43, 49, 51, 57, 59).

[61] I. D. Coope and C. J. Price. "Positive bases in numerical optimization". In: *Comput. Optim. Appl.* 21 (2002) (cit. on p. 90).

[62] Y. Cui and J.-S. Pang. *Modern Nonconvex Nondifferentiable Optimization*. Philadelphia, PA: SIAM, 2021, pp. 523–568 (cit. on p. 31).

[63] A. L. Custódio, Y. Diouane, R. Garmanjani, and E. Riccietti. "Worst-case complexity bounds of directional direct-search methods for multiobjective optimization". In: *J. Optim. Theory Appl.* 188 (2021), pp. 73–93 (cit. on pp. 22, 46, 56, 57, 69).

[64] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. "Direct multisearch for multiobjective optimization". In: *SIAM J. Optim.* 21 (2011), pp. 1109–1140 (cit. on pp. 4, 40, 44–46, 48, 49, 53, 55, 58–60, 62–64, 68, 70, 83, 109, 111).

[65] A. L. Custódio, H. Rocha, and L. N. Vicente. "Incorporating minimum Frobenius norm models in direct search". In: *Comput. Optim. Appl.* 46 (2010), pp. 265–278 (cit. on pp. 5, 71, 99, 100).

[66] A. L. Custódio and L. N. Vicente. "Using sampling and simplex derivatives in pattern search methods". In: *SIAM J. Optim.* 18 (2007), pp. 537–555 (cit. on pp. 5, 71, 99, 100).

[67] N. O. Da Cunha and E. Polak. "Constrained minimization under vector-valued criteria in finite dimensional spaces". In: *J. Math. Anal. Appl.* 19 (1967), pp. 103–124 (cit. on p. 23).

[68] I. Das and J. E. Dennis Jr. "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems". In: *SIAM J. Optim.* 8 (1998), pp. 631–657 (cit. on pp. 40, 44).

[69] R. De Leone, M. Gaudioso, and L. Grippo. "Stopping criteria for linesearch methods without derivatives". In: *Math. Program.* 30 (1984), pp. 285–300 (cit. on p. 88).

[70] K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001 (cit. on pp. 2, 40).

[71] K. Deb and H. Jain. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints". In: *IEEE Trans. on Evolut. Comput.* 18 (2014), pp. 577–601 (cit. on p. 77).

[72] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE T. Evolut. Comput.* 6 (2002), pp. 182–197 (cit. on p. 49).

[73] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. "Scalable multi-objective optimization test problems". In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. Vol. 1. 2002, pp. 825–830 (cit. on p. 75).

[74] S. Deshpande, L. T. Watson, and R. A. Canfield. "Multiobjective optimization using an adaptive weighting scheme". In: *Optim. Methods Softw.* 31 (2016), pp. 110–133 (cit. on pp. 44, 49).

[75] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Math. Program.* 91 (2002), pp. 201–213 (cit. on p. 39).

[76] L. M. Graña Drummond and A. N. Iusem. "A projected gradient method for vector optimization problems". In: *Comput. Optim. Appl.* 28 (2004), pp. 5–29 (cit. on p. 40).

[77] L. M. Graña Drummond, F. M. P. Raupp, and B. F. Svaiter. "A quadratically convergent Newton method for vector optimization". In: *Optimization* 63 (2014), pp. 661–677 (cit. on p. 40).

[78] L. M. Graña Drummond and B. F. Svaiter. "A steepest descent method for vector optimization". In: *J. Comput. Appl. Math.* 175 (2005), pp. 395–414 (cit. on p. 40).

[79] K. J. Dzahini, F. Rinaldi, C. W. Royer, and D. Zeffiro. *Revisiting theoretical guarantees of direct-search methods*. 2024. arXiv: 2403.05322 [math.OC] (cit. on p. 37).

[80] G. Eichfelder. *Adaptive Scalarization Methods in Multiobjective Optimization*. Vector Optimization. Heidelberg, Germany: Springer, 2008 (cit. on pp. 2, 40, 44).

[81] M. T. M. Emmerich and A. H. Deutz. "A tutorial on multiobjective optimization: fundamentals and evolutionary methods". In: *Nat. Computing* 17 (2018), pp. 585–609 (cit. on pp. 63, 68).

[82] R. Escalante and M. Raydan. *Alternating Projection Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2011 (cit. on p. 30).

[83] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi. "A linesearch-based derivative-free approach for nonsmooth constrained optimization". In: *SIAM J. Optim.* 24 (2014), pp. 959–992 (cit. on p. 18).

[84] N. S. Fazzio and M. L. Schuverdt. "Convergence analysis of a nonmonotone projected gradient method for multiobjective optimization problems". In: *Optim. Lett.* 13 (2019), pp. 1365–1379 (cit. on p. 40).

[85] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Vol. 4. Classics in Applied Mathematics. Philadelphia: Society for Industrial and Applied Mathematics, 1990 (cit. on p. 31).

[86] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter. "Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming". In: *SIAM J. Optim.* 13 (2002), pp. 635–659 (cit. on p. 32).

[87] R. Fletcher and S. Leyffer. "Nonlinear programming without a penalty function". In: *Math. Program.* 91 (2002), pp. 239–269 (cit. on pp. 32, 34, 47, 51).

[88] R. Fletcher, S. Leyffer, and Ph. L. Toint. "A brief history of filter methods". In: *SIAG/OPT Views-and-News* 18 (2007), pp. 2–12 (cit. on p. 33).

[89] J. Fliege, L. M. Graña Drummond, and B. F. Svaiter. "Newton's method for multiobjective optimization". In: *SIAM J. Optim.* 20 (2009), pp. 602–626 (cit. on pp. 21, 40, 42).

[90] J. Fliege and B. F. Svaiter. "Steepest descent methods for multicriteria optimization". In: *Math. Meth. Oper. Res.* 51 (2000), pp. 479–494 (cit. on pp. 21, 22, 40).

[91] J. Fliege and A. I. F. Vaz. "A method for constrained multiobjective optimization based on SQP techniques". In: *SIAM J. Optim.* 26 (2016), pp. 2091–2119 (cit. on p. 43).

[92] A. Forsgren, Ph. E. Gill, and M. H. Wright. "Interior methods for nonlinear optimization". In: *SIAM Rev.* 44 (2002), pp. 525–597 (cit. on p. 32).

[93] E. H. Fukuda, L. M. G. Drummond, and F. M. P. Raupp. "A barrier-type method for multiobjective optimization". In: *Optimization* 69 (2020), pp. 2471–2487 (cit. on p. 107).

[94] E. H. Fukuda and L. M. Graña Drummond. "Inexact projected gradient method for vector optimization". In: *Comput. Optim. Appl.* 54 (2013), pp. 473–493 (cit. on p. 40).

[95] E. H. Fukuda and L. M. Graña Drummond. "On the convergence of the projected gradient method for vector optimization". In: *Optimization* 60 (2011), pp. 1009–1021 (cit. on p. 40).

[96] J. L. Morales G. Fasano and J. Nocedal. "On the geometry phase in model-based algorithms for derivative-free optimization". In: *Optim. Methods Softw.* 24 (2009), pp. 145–154 (cit. on p. 38).

[97] J. D. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. 5th ed. New York: CRC Press, 2010 (cit. on p. 73).

[98] Ph. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019 (cit. on pp. 28, 30).

[99] R. H. Gómez and C. A. Coello Coello. "Improved metaheuristic based on the R2 indicator for many-objective optimization". In: Association for Computing Machinery, Inc, 2015-07, pp. 679–686 (cit. on p. 77).

[100] M. Gonçalves, F. Lima, and L. Prudente. "A study of Liu-Storey conjugate gradient methods for vector optimization". In: *Appl. Math. Comput.* 425 (2022), p. 127099 (cit. on p. 40).

[101] M. L. N. Gonçalves, F. S. Lima, and L. F. Prudente. "Globally convergent Newton-type methods for multiobjective optimization". In: *Comput. Optim. Appl.* 83 (2022), pp. 403–434 (cit. on p. 40).

[102] M. L. N. Gonçalves and L. F. Prudente. "On the extension of the Hager–Zhang conjugate gradient method for vector optimization". In: *Comput. Optim. Appl.* 76 (2020), pp. 889–916 (cit. on p. 40).

[103] C. Gonzaga, E. Karas, and M. Vanti. "A globally convergent filter method for nonlinear programming". In: *SIAM J. Optim.* 14 (2004), pp. 646–669 (cit. on p. 32).

[104] F. J. Gould and Jon W. Tolle. "A necessary and sufficient qualification for constrained optimization". In: *SIAM J. Appl. Math.* 20 (1971), pp. 164–172 (cit. on p. 11).

[105] N . Gould and J. Scott. "A note on performance profiles for benchmarking software". In: *ACM Trans. Math. Softw.* 43 (2016) (cit. on p. 79).

[106] N. I. M. Gould, D. Orban, and Ph. L. Toint. "CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization". In: *Comput. Optim. Appl.* 60 (2015), pp. 545–557 (cit. on p. 101).

[107] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. "Direct search based on probabilistic descent". In: *SIAM J. Optim.* 25 (2015), pp. 1515–1541 (cit. on p. 68).

[108] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang. "Direct search based on probabilistic feasible descent for bound and linearly constrained problems". In: *Comput. Optim. Appl.* 72 (2019), pp. 525–559 (cit. on p. 90).

[109] I. Griva, S. G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2009 (cit. on pp. 9, 27, 28).

[110] A. Guerraggio and D. T. Luc. "Optimality conditions for $C^{1,1}$ vector optimization problems". In: *J. Optim. Theory Appl.* 109 (2001), pp. 615–629 (cit. on p. 21).

[111] J. Handl, D. B. Kell, and J. Knowles. "Multiobjective optimization in bioinformatics and computational biology". In: *IEEE/ACM Trans. Computational Biology and Bioinformatics* 4 (2007), pp. 279–292 (cit. on p. 40).

[112] W. Hare and G. Jarry-Bolduc. "A deterministic algorithm to compute the cosine measure of a finite positive spanning set". In: *Optim. Lett.* 14 (2020) (cit. on p. 90).

[113] W. Hare, G. Jarry-Bolduc, and C. Planiden. "Nicely structured positive bases with maximal cosine measure". In: *Optim. Lett.* 17 (2023) (cit. on p. 90).

[114] M. R. Hestenes. "Multiplier and gradient methods". In: *J. Optim. Theory Appl.* 4 (1969), pp. 303–320 (cit. on p. 32).

[115] J. B. Hiriart-Urruty. "On optimality conditions in nondifferentiable programming". In: *Math. Program.* 14 (1978), pp. 73–86 (cit. on p. 13).

[116] S. Huband, L. Barone, L. While, and P. Hingston. "A scalable multi-objective test problem toolkit". In: *Evolutionary Multi-Criterion Optimization*. Ed. by C. A. Coello Coello, A. H. Aguirre, and E. Zitzler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 280–295 (cit. on p. 75).

[117] J. Jahn. *Introduction to the Theory of Nonlinear Optimization*. 4th ed. Springer Cham, 2020 (cit. on pp. 8, 15).

[118] Y. Jin and B. Sendhoff. "Pareto-based multiobjective machine learning: An overview and case studies". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38 (2008), pp. 397–415 (cit. on p. 40).

[119] N. Karmitsa. *Test Problems for Large-scale Nonsmooth Minimization*. Reports of the Department of Mathematical Information Technology. Finland: University of Jyväskylä, 2007 (cit. on pp. 62–64, 66).

[120] C. T. Kelley. *Implicit Filtering*. Software Environments and Tools. Philadelphia: SIAM, 2011 (cit. on p. 47).

[121] T. G. Kolda, R. M. Lewis, and V. Torczon. "Optimization by direct search: New perspectives on some classical and modern methods". In: *SIAM Rev.* 45 (2003), pp. 385–482 (cit. on pp. 34, 36, 56, 88, 90).

[122] T. G. Kolda, R. M. Lewis, and V. Torczon. "Stationarity results for generating set search for linearly constrained optimization". In: *SIAM J. Optim.* 17 (2006), pp. 943–968 (cit. on pp. 33, 90, 91).

[123] M. Lapucci. "Convergence and complexity guarantees for a wide class of descent algorithms in nonconvex multi-objective optimization". In: *Oper. Res. Lett.* 54 (2024), p. 107115 (cit. on p. 41).

[124] M. Lapucci and P. Mansueto. "Improved front steepest descent for multi-objective optimization". In: *Oper. Res. Lett.* 51 (2023), pp. 242–247 (cit. on p. 40).

[125] M. Lapucci, P. Mansueto, and D. Pucci. *Effective front-descent algorithms with convergence guarantees*. 2024. arXiv: 2405.08450 [math.OC] (cit. on pp. 40, 41).

[126] J. Larson, M. Menickelly, and S. M. Wild. "Derivative-free optimization methods". In: *Acta Numer.* 28 (2019), pp. 287–404 (cit. on pp. 3, 49).

[127] J.B. Lasserre. "A globally convergent algorithm for exact penalty functions". In: *European J. Oper. Res.* 7 (1981), pp. 389–395 (cit. on p. 31).

[128] R. M. Lewis and V. Torczon. "A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds". In: *SIAM J. Optim.* 12 (2002), pp. 1075–1089 (cit. on p. 33).

[129] R. M. Lewis and V. Torczon. "Pattern search algorithms for bound constrained minimization". In: *SIAM J. Optim.* 9 (1999), pp. 1082–1099 (cit. on p. 33).

[130] R. M. Lewis and V. Torczon. "Pattern search methods for linearly constrained minimization". In: *SIAM J. Optim.* 10 (2000), pp. 917–941 (cit. on pp. 33, 91, 103).

[131] B. Li, J. Li, K. Tang, and X. Yao. "Many-objective evolutionary algorithms: A survey". In: *ACM Comput. Surv.* 48 (2015) (cit. on p. 68).

[132] K. Li, K. Deb, Q. Zhang, and S. Kwong. "An evolutionary many-objective optimization algorithm based on dominance and decomposition". In: *IEEE Trans. on Evolut. Comput.* 19 (2015-10), pp. 694–716 (cit. on p. 77).

[133] G. Liuzzi and S. Lucidi. "A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function". In: *SIAM J. Optim.* 20 (2009), pp. 1–29 (cit. on p. 34).

[134] G. Liuzzi, S. Lucidi, and F. Rinaldi. "A derivative-free approach to constrained multiobjective nonsmooth optimization". In: *SIAM J. Optim.* 26 (2016), pp. 2744–2774 (cit. on pp. 24, 25, 47, 49, 62, 64, 66, 84).

[135] G. Liuzzi, S. Lucidi, and M. Sciandrone. "A derivative-free algorithm for linearly constrained finite minimax problems". In: *SIAM J. Optim.* 16 (2006), pp. 1054–1075 (cit. on p. 91).

[136] G. Liuzzi, S. Lucidi, and M. Sciandrone. "Sequential penalty derivative-free methods for nonlinear constrained optimization". In: *SIAM J. Optim.* 20 (2010), pp. 2614–2635 (cit. on p. 34).

[137] João M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[138] S. Lucidi, M. Sciandrone, and P. Tseng. "Objective-derivative-free methods for constrained optimization". In: *Math. Program.* 92 (2002), pp. 37–59 (cit. on pp. 33, 91, 103).

[139] Z. Ma and Y. Wang. "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons". In: *IEEE Trans. on Evolut. Comput.* 23 (2019), pp. 972–986 (cit. on pp. 111, 113, 114).

[140] N. Mahdavi-Amiri and F. S. Sadaghiani. "A superlinearly convergent nonmonotone quasi-Newton method for unconstrained multiobjective optimization". In: *Optim. Methods Softw.* 35 (2020), pp. 1223–1247 (cit. on p. 40).

[141] O. L. Mangasarian. *Nonlinear Programming*. Society for Industrial and Applied Mathematics, 1994 (cit. on p. 11).

[142] I. Maruşciac. "On Fritz John type optimality criterion in multi-objective optimization". In: *L'Analyse Numérique et la Théorie de l'Approximation* 11 (1982), pp. 109–114 (cit. on p. 23).

[143] J. H. May. "Linearly Constrained Nonlinear Programming: A Solution Method that Does Not Require Analytic Derivatives". PhD thesis. Yale University, 1974 (cit. on p. 33).

[144] K. Miettinen. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science. New York, USA: Springer US, 1998 (cit. on pp. 2, 3, 20, 40).

[145] A. Mohammadi. "Trust-Region Methods for Multiobjective Derivative-free Optimization". PhD thesis. Lisbon, Portugal: NOVA University of Lisbon, 2024 (cit. on pp. 44, 49).

[146] A. Mohammadi and A. L. Custódio. "A trust-region approach for computing Pareto fronts in multiobjective optimization". In: *Comput. Optim. Appl.* 87 (2024), pp. 149–179 (cit. on pp. 43, 44).

[147] J. J. Moré and S. M. Wild. "Benchmarking derivative-free optimization algorithms". In: *SIAM J. Optim.* 20 (2009), pp. 172–191 (cit. on p. 39).

[148] V. Morovati, H. Basirzadeh, and L. Pourkarimi. "Quasi-Newton methods for multiobjective optimization problems". In: *4OR-Q J. Oper. Res.* 16 (2017), pp. 261–294 (cit. on p. 40).

[149] H. Moulin and F. Fogelman-Soulié. *La Convexité dans les Mathématiques de la Décision: Optimisation et Théorie Micro-Économique*. French. Méthodes. Hermann, 1979, p. 294 (cit. on p. 19).

[150] J. A. Nelder and R. Mead. "A simplex method for function minimization". In: *Comput. J.* 7 (1965), pp. 308–313 (cit. on p. 33).

[151] S. Nobakhtian. "Infine functions and nonsmooth multiobjective optimization problems". In: *Comput. Math. Appl.* 51 (2006), pp. 1385–1394 (cit. on p. 25).

[152] J. Nocedal and S. J. Wright. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. New York: Springer New York, 2006 (cit. on pp. 10, 27, 28, 30).

[153] J. Peng, J. Zhang, and J. Yao. "A novel hybrid conjugate gradient method for multiobjective optimization problems". In: *Optimization* (2024), pp. 1–20. DOI: 10.1080/02331934.2024.2373903 (cit. on p. 40).

[154] L. R. Lucambio Pérez and L. F. Prudente. "Nonlinear conjugate gradient methods for vector optimization". In: *SIAM J. Optim.* 28 (2018), pp. 2690–2720 (cit. on p. 40).

[155] P. S. Potrebko, J. Fiege, M. Biagioli, and J. Poleszczuk. "Investigating multi-objective fluence and beam orientation IMRT optimization". In: *Phys. Med. Biol.* 62 (2017), pp. 5228–5244 (cit. on p. 40).

[156] Ž. Povalej. "Quasi-Newton's method for multiobjective optimization". In: *J. Comput. Appl. Math.* 255 (2014), pp. 765–777 (cit. on p. 40).

[157] M. J. D. Powell. "A direct search optimization method that models the objective and constraint functions by linear interpolation". In: *Advances in Optimization and Numerical Analysis*. Ed. by S. Gomez and J.-P. Hennart. Vol. 275. Mathematics and Its Applications. Dordrecht, Netherlands: Springer, 1994, pp. 51–67 (cit. on p. 38).

[158] M. J. D. Powell. "A method for nonlinear constraints in minimization problems". In: *Optimization* (1969). Ed. by R. Fletcher, pp. 283–298 (cit. on p. 32).

[159] M. J. D. Powell. "Least Frobenius norm updating of quadratic models that satisfy interpolation conditions". In: *Math. Program.* B 100 (2004), pp. 183–215 (cit. on p. 38).

[160] M. J. D. Powell. "On the use of quadratic models in unconstrained minimization without derivatives". In: *Optim. Methods Softw.* 19 (2004), pp. 399–411 (cit. on p. 38).

[161] M. J. D. Powell. "On trust region methods for unconstrained minimization without derivatives". In: *Math. Program.* B 97 (2003), pp. 605–623 (cit. on p. 38).

[162] M. J. D. Powell. *The BOBYQA algorithm for bound constrained optimization without derivatives*. Tech. rep. DAMTP 2009/NA06. Cambridge, UK: Department of Applied Mathematics and Theoretical Physics, University of Cambridge, 2009 (cit. on p. 38).

[163] M. J. D. Powell. "The NEWUOA software for unconstrained optimization without derivatives". In: *Large-Scale Nonlinear Optimization*. Ed. by G. Di Pillo and M. Roma. Vol. 83. Nonconvex Optimization and Its Applications. Boston, MA, USA: Springer, 2006, pp. 255–297 (cit. on p. 38).

[164] M. J. D. Powell. "UOBYQA: unconstrained optimization by quadratic approximation". In: *Math. Program.* 92 (2002), pp. 555–582 (cit. on p. 38).

[165] L. F. Prudente and D. R. Souza. "A quasi-Newton method with Wolfe line searches for multiobjective optimization". In: *J. Optim. Theory Appl.* 194 (2022), pp. 1107–1140 (cit. on p. 40).

[166] L. F. Prudente and D. R. Souza. "Global convergence of a BFGS-type algorithm for nonconvex multiobjective optimization problems". In: *Comput. Optim. Appl.* 88 (2024) (cit. on p. 40).

[167] S. Qu, M. Goh, and F. T. Chan. "Quasi-Newton methods for solving multiobjective optimization". In: *Oper. Res. Lett.* 39 (2011), pp. 397–399 (cit. on p. 40).

[168] S. Qu, M. Goh, and B. Liang. "Trust region methods for solving multiobjective optimisation". In: *Optim. Methods Softw.* 28 (2013), pp. 796–811 (cit. on pp. 42, 43).

[169] S. Qu, C. Liu, M. Goh, Y. Li, and Y. Ji. "Nonsmooth multiobjective programming with quasi-Newton methods". In: *European J. Oper. Res.* 235 (2014), pp. 503–510 (cit. on p. 40).

[170] V. A. Ramirez and G. N. Sottosanto. "Nonmonotone trust region algorithm for solving the unconstrained multiobjective optimization problems". In: *Comput. Optim. Appl.* 81 (2022), pp. 769–788 (cit. on p. 43).

[171] L. Roberts and C. W. Royer. "Direct search based on probabilistic descent in reduced spaces". In: *SIAM J. Optim.* 33 (2023), pp. 3057–3082 (cit. on pp. 4, 68, 74, 76).

[172] F. Rosso, V. Ciancio, J. Dell'Olmo, and F. Salata. "Multi-objective optimization of building retrofit in the Mediterranean climate by means of genetic algorithm application". In: *Energy and Buildings* 216 (2020), p. 109945 (cit. on p. 40).

[173] W. Rudin. *Functional Analysis*. 2nd. International Series in Pure and Applied Mathematics. New York: McGraw-Hill, Inc., 1991 (cit. on p. 144).

[174] J.-H. Ryu and S. Kim. "A derivative-free trust-region method for biobjective optimization". In: *SIAM J. Optim.* 24 (2014), pp. 334–362 (cit. on pp. 44, 49).

[175] K. Scheinberg and Ph. L. Toint. "Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization". In: *SIAM J. Optim.* 20 (2010), pp. 3512–3532 (cit. on p. 38).

[176] E. J. Silva and A. L. Custódio. "An inexact restoration direct multisearch filter approach to multiobjective constrained derivative-free optimization". In: *Optim. Methods Softw.* (2024), pp. 1–27. DOI: 10.1080/10556788.2024.2412646 (cit. on pp. 4, 51).

[177] C. Singh. "Optimality conditions in multiobjective differentiable programming". In: *J. Optim. Theory Appl.* 53 (1987), pp. 115–123 (cit. on p. 23).

[178] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino. "GA-based decision support system for multicriteria optimization". In: *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*. Vol. 2. 1995, pp. 1556–1561 (cit. on p. 111).

[179] S. Tavares, C. P. Brás, A. L. Custódio, V. Duarte, and P. Medeiros. "Parallel strategies for Direct Multisearch". In: *Numer. Algor.* 92 (2023), pp. 1757–1788 (cit. on pp. 83, 120).

[180] J. Thomann and G. Eichfelder. "A trust-region algorithm for heterogeneous multiobjective optimization". In: *SIAM J. Optim.* 29 (2019), pp. 1017–1047 (cit. on p. 47).

[181] Y. Tian, R. Cheng, X. Zhang, and Y. Jin. "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization". In: *IEEE Comput. Intell. Mag.* 12 (2017), pp. 73–87 (cit. on p. 77).

[182] V. Torczon. "On the convergence of pattern search algorithms". In: *SIAM J. Optim.* 7 (1997), pp. 1–25 (cit. on p. 33).

[183] L. N. Vicente and A. L. Custódio. "Analysis of direct searches for discontinuous functions". In: *Math. Program.* 133 (2012), pp. 299–325 (cit. on p. 59).

[184] J. Wang, Y. Hu, C. K. Wai Yu, C. Li, and X. Yang. "Extended Newton methods for multiobjective optimization: majorizing function technique and convergence analysis". In: *SIAM J. Optim.* 29 (2019), pp. 2388–2421 (cit. on p. 40).

[185] M. M. Wiecek, M. Ehrgott, and A. Engau. "Continuous multiobjective programming". In: *Multiple Criteria Decision Analysis*. New York, USA: Springer, 2016, pp. 739–815 (cit. on pp. 3, 20).

[186] S. Yang, M. Li, X. Liu, and J. Zheng. "A grid-based evolutionary algorithm for many-objective optimization". In: *IEEE Trans. on Evolut. Comput.* 17 (2013), pp. 721–736 (cit. on p. 77).

[187] X. Zhang, Y. Tian, and Y. Jin. "A knee point-driven evolutionary algorithm for many-objective optimization". In: *IEEE Trans. on Evolut. Comput.* 19 (2015-12), pp. 761–776 (cit. on p. 77).

[188] Z. Zhang. *PRIMA: Reference implementation for Powell's methods with modernization and amelioration*. available at http://www.libprima.net,DOI:10.5281/zenodo.8052654. 2023 (cit. on p. 38).

[189] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca. "Performance assessment of multiobjective optimizers: An analysis and review". In: *IEEE T. Evolut. Comput.* 7 (2003), pp. 117–132 (cit. on pp. 48, 56).

A

Individual Comparisons - DMS-FILTER-IR and other solvers

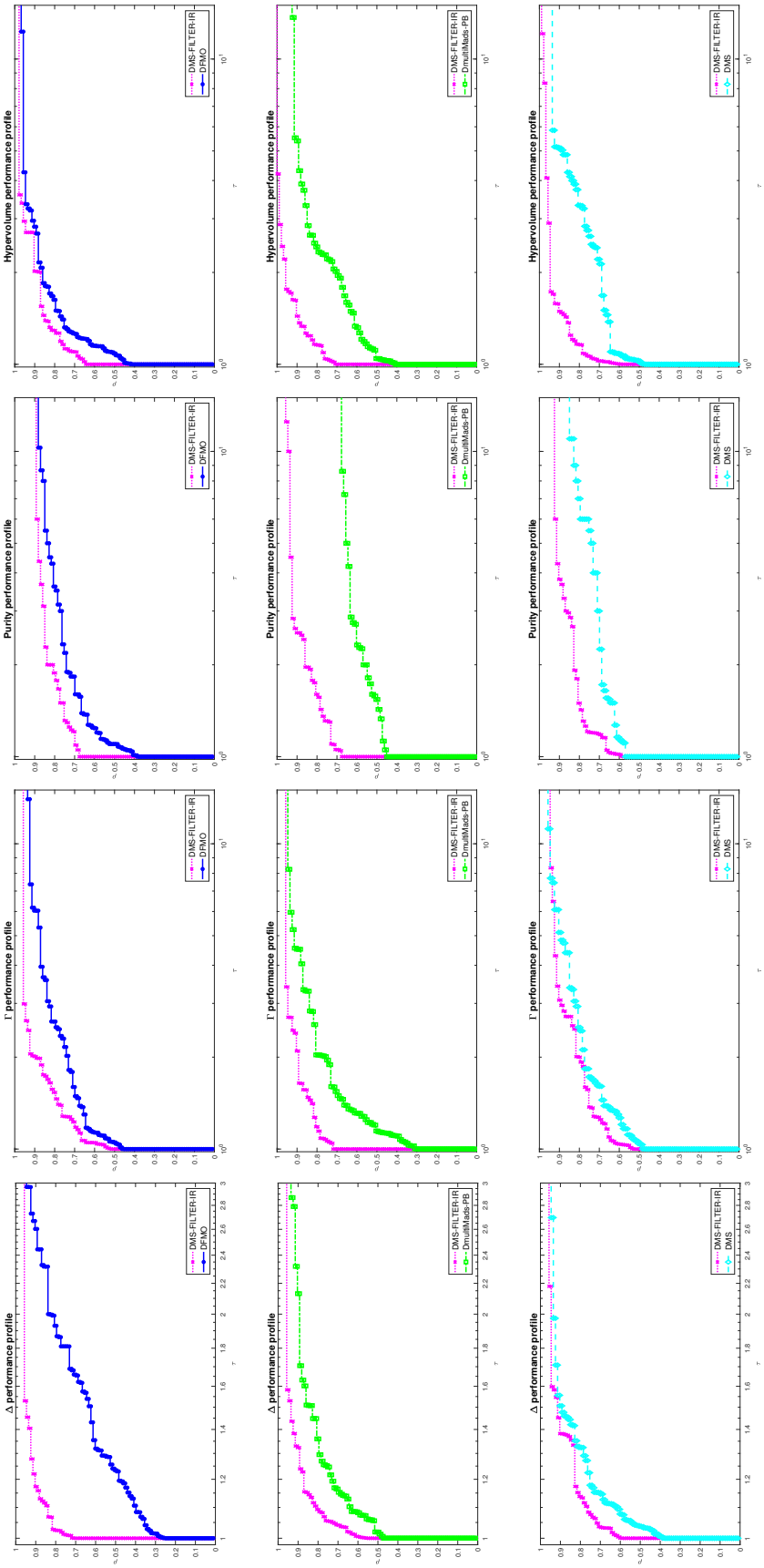


Figure A.1: Individual comparison between DMS-FILTER-IR and DFMO, DMultiMADS-PB, and DMS based on performance profiles for a maximum budget of 500 function evaluations.

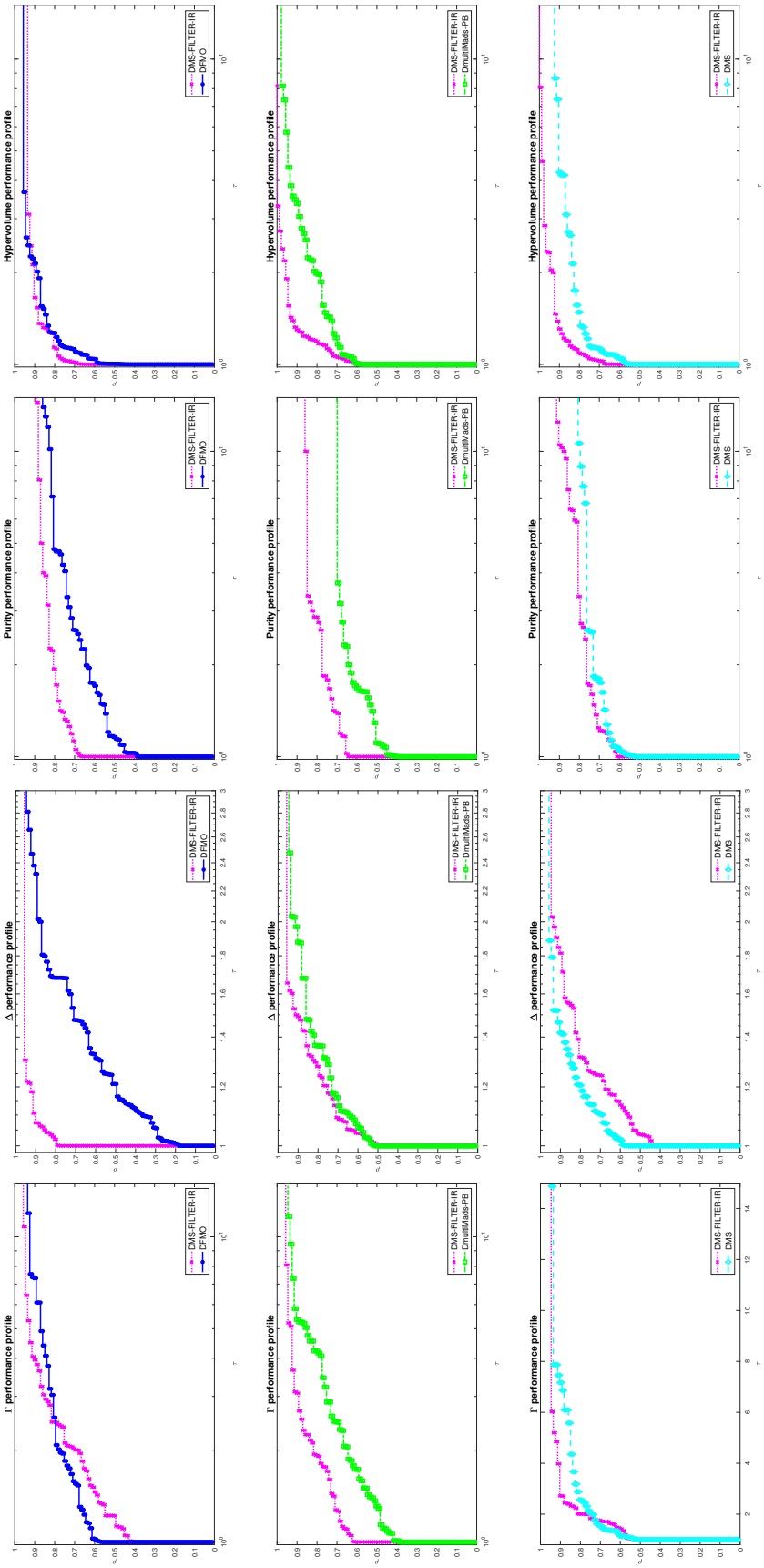


Figure A.2: Individual comparison between DMS-FILTER-IR and DFMO, DMultiMADS-PB, and DMS based on performance profiles for a maximum budget of 5000 function evaluations.

B

Individual Comparisons - DMS-Reduction and other solvers

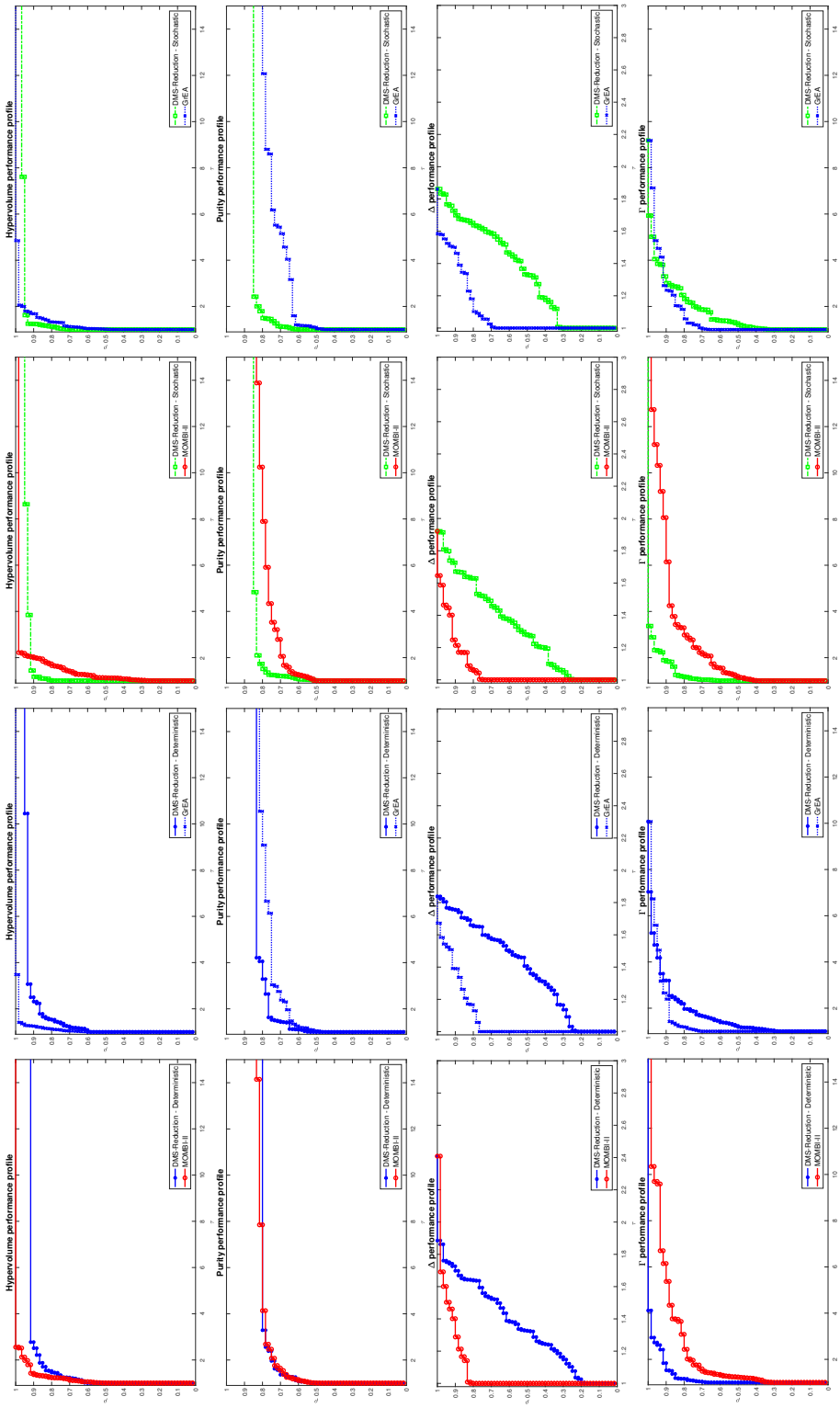


Figure B.1: Individual comparison between the Deterministic and Stochastic versions of DMS-Reduction, employing a simple decrease to accept new nondominated points, MOMBI-II, and GrEA based on performance profiles for a maximum budget of 5000 function evaluations.

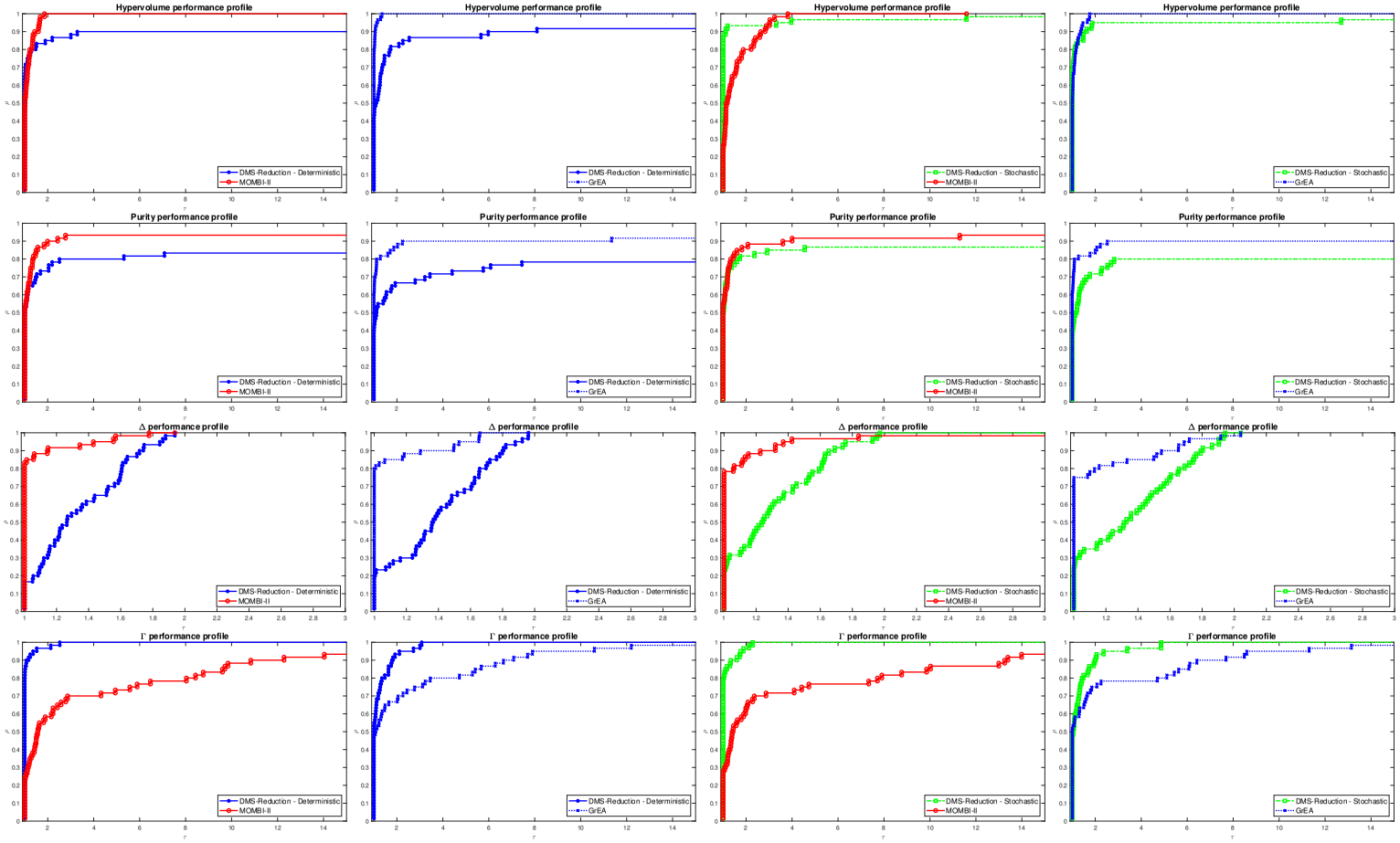


Figure B.2: Individual comparison between the Deterministic and Stochastic versions of DMS-Reduction, employing a simple decrease to accept new nondominated points, MOMBI-II, and GrEA based on performance profiles for a maximum budget of 20000 function evaluations.

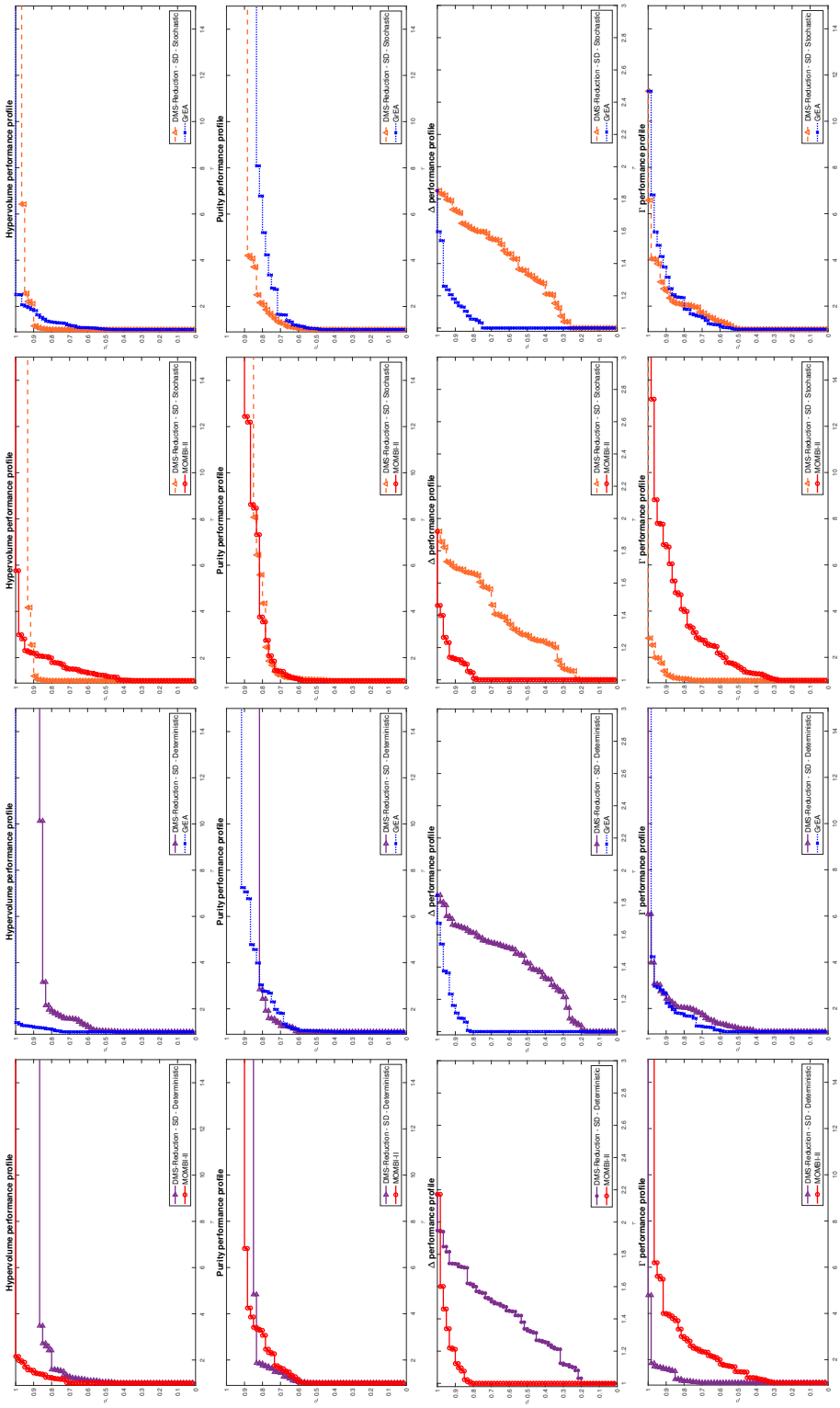


Figure B.3: Individual comparison between the Deterministic and Stochastic versions of DMS-Reduction, employing a sufficient decrease to accept new nondominated points, MOMBI-II, and GrEA based on performance profiles for a maximum budget of 5000 function evaluations.

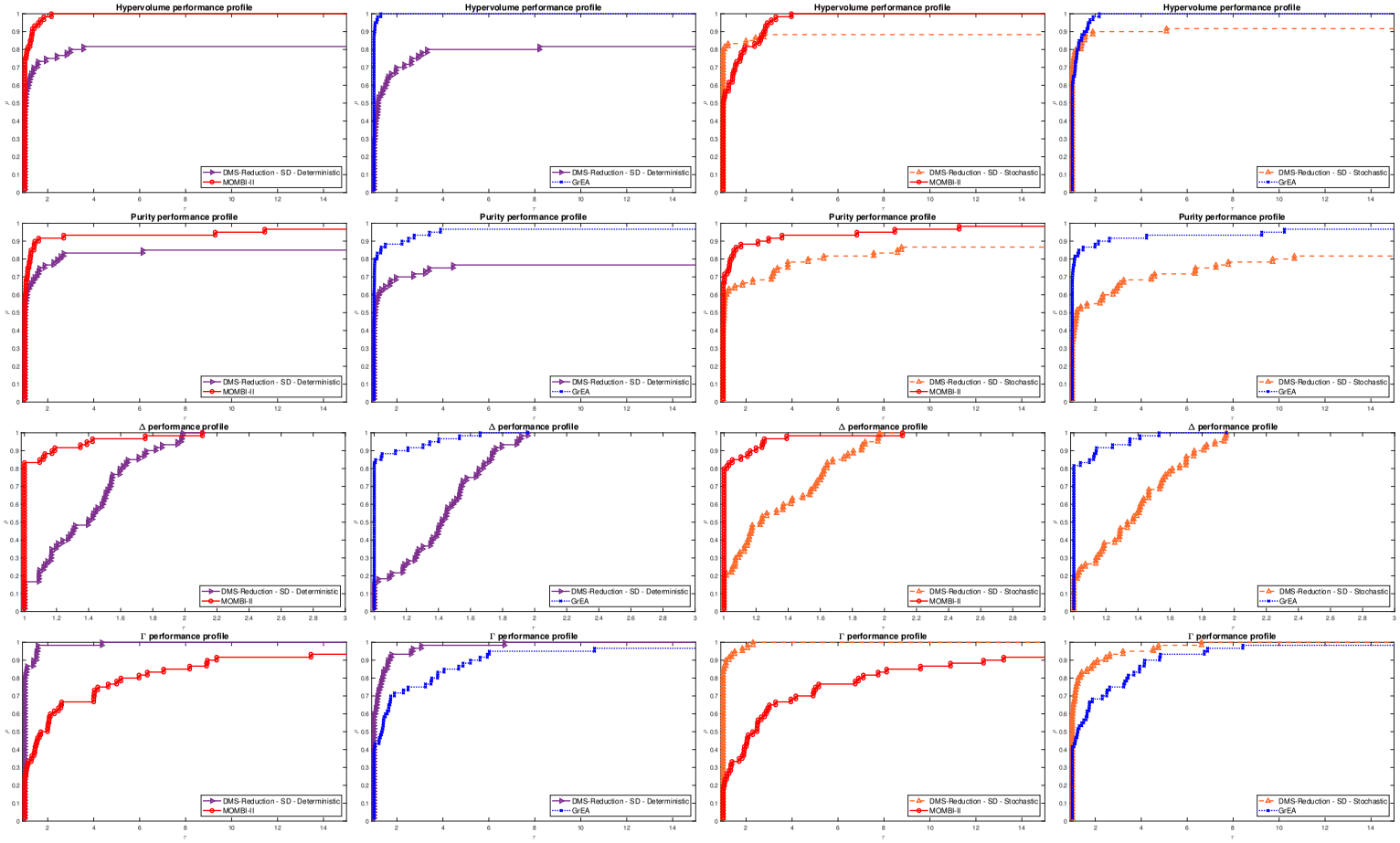


Figure B.4: Individual comparison between the Deterministic and Stochastic versions of DMS-Reduction, employing a sufficient decrease to accept new nondominated points, MOMBI-II, and GrEA based on performance profiles for a maximum budget of 20000 function evaluations.

C

Technical Results and Boundedness of Lagrange Multipliers for LOG-DS

In this appendix, we present technical results involving inequalities for real numbers and exponents, a property of sequences of scalars, and, finally, the proof of the boundedness of the Lagrange multipliers required in the proofs of Chapter 6.

The first lemma provides a bound for the sum of powers of positive real numbers.

Lemma C.0.1. *[173, Equation (2), p. 37] Let $a \in \mathbb{R}_+$, $b \in \mathbb{R}_+$, $a + b > 0$ and $p \in [0, 1]$. It results*

$$(a + b)^p \leq a^p + b^p.$$

The next result shows bounds concerning the difference of powers of absolute values and maximum functions, which are fundamental for proving the boundedness of the Lagrange multipliers.

Lemma C.0.2. *Let $a \in \mathbb{R}$, $b \in \mathbb{R}$ and $p \in [0, 1]$. The following inequalities hold*

$$||a|^p - |b|^p| \leq |a - b|^p, \tag{C.1}$$

and

$$|\max\{a, 0\}^p - \max\{b, 0\}^p| \leq |a - b|^p. \tag{C.2}$$

Proof. To prove (C.1) consider

$$\begin{aligned} |a|^p &= |a - b + b|^p \\ \text{(Triangular inequality)} &\leq (|a - b| + |b|)^p \\ \text{(Lemma C.0.1)} &\leq |a - b|^p + |b|^p \end{aligned} \tag{C.3}$$

On the other hand,

$$\begin{aligned} |b|^p &= |b - a + a|^p \\ \text{(Triangular inequality)} &\leq (|b - a| + |a|)^p \\ \text{(Lemma C.0.1)} &\leq |a - b|^p + |a|^p \end{aligned} \tag{C.4}$$

Then, from inequalities (C.3) and (C.4) we derive inequality (C.1).
Now, to prove (C.2), 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 (C.1) we conclude that:

$$|\max\{a,0\}^p - \max\{b,0\}^p| = |a^p - b^p| = ||a|^p - |b|^p| \leq |a-b|^p.$$

□

Now, we present a result regarding a property of sequences of scalars, which is used to prove the boundedness of the Lagrange multipliers.

Lemma C.0.3. *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:

$$\lim_{\substack{k\to+\infty,\\ k\in K}} \frac{a_k^i}{|a_k^s|} = z_i, \quad |z_i| < +\infty, \quad i = 1,\dots,\ell, \tag{C.5}$$

i.e., all the sequences $\left\{\dfrac{a_k^i}{|a_k^s|}\right\}_{k\in K}$ are bounded.

Proof. Point (i) directly follows from the properties of convergent sequences.

Then, let us now assume that at least one sequence is not convergent to zero. If this is the case, we can reorder the sequences in such a way that:

- $\lim_{k\to+\infty} a_k^1 = \lim_{k\to+\infty} a_k^2 = \cdots = \lim_{k\to+\infty} a_k^{r-1} = 0$;
- $\{a_k^i\}$, $i = r,\dots,\ell$, are not convergent to zero.

We now prove that an index $s \in \{r,\dots,\ell\}$ and an infinite index set $\hat{K}$ exist such that

$$\lim_{\substack{k\to+\infty,\\ k\in\hat{K}}} \frac{a_k^i}{|a_k^s|} = z_i, \quad |z_i| < +\infty, \quad i = r,\dots,\ell.$$

This is (obviously) true when $\ell = r$. Indeed, when $\ell = r$, there is K_r such that

$$\lim_{\substack{k\to+\infty,\\ k\in K_r}} \frac{a_k^r}{|a_k^r|} = \pm 1.$$

Furthermore, we have

$$\lim_{\substack{k\to+\infty,\\ k\in K_r}} \frac{a_k^i}{|a_k^r|} = 0, \quad i = 1, \dots, r-1.$$

The thesis follows by choosing $s = r$ and $K = \hat{K} = K_r$.

Now, we prove the thesis by induction on ℓ. Then, we have sequences $\{a_k^i\}$, $i = 1, \dots, \ell - 1$ such that:

- $\lim_{k\to+\infty} a_k^1 = \lim_{k\to+\infty} a_k^2 = \cdots = \lim_{k\to+\infty} a_k^{r-1} = 0$;
- $\{a_k^i\}$, $i = r, \dots, \ell - 1$, are not convergent to zero,

and an index $\hat{\imath} \in \{r, \dots, \ell - 1\}$ and an infinite index set $\hat{K}$ exist such that

$$\lim_{\substack{k\to+\infty,\\ k\in \hat{K}}} \frac{a_k^i}{|a_k^{\hat{\imath}}|} = z_i, \quad |z_i| < +\infty, \quad i = r, \dots, \ell - 1.$$

1. If $\lim_{k\to+\infty} a_k^\ell = 0$, then we have

$$\lim_{\substack{k\to+\infty,\\ k\in \hat{K}}} \frac{a_k^j}{|a_k^{\hat{\imath}}|} = z_j, \quad j = 1, \dots, \ell - 1$$
$$\lim_{\substack{k\to+\infty,\\ k\in \hat{K}}} \frac{a_k^\ell}{|a_k^{\hat{\imath}}|} = 0;$$

   and the thesis follows by choosing $s = \hat{\imath}$ and $K = \hat{K}$.

2. Suppose now that $\{a_k^\ell\}$ is not convergent to zero. In this situation, two subcases can occur:

   a) the sequence $\left\{\frac{a_k^\ell}{a_k^{\hat{\imath}}}\right\}_{k\in\hat{K}}$ is bounded;

   b) an infinite index set $K_1 \subseteq \hat{K}$ exists such that $\left\{\frac{a_k^\ell}{a_k^{\hat{\imath}}}\right\}_{k\in K_1}$ is unbounded.

   In the first case, an infinite index set $K_2 \subseteq \hat{K}$ exists such that $\left\{\frac{a_k^\ell}{a_k^{\hat{\imath}}}\right\}_{k\in K_2}$ is convergent. Then, the thesis follows by taking $s = \hat{\imath}$ and $K = K_2$.

   In the second case, we have

$$\lim_{\substack{k\to+\infty,\\ k\in K_1}} \frac{a_k^{\hat{\imath}}}{a_k^\ell} = 0.$$

Furthermore, for every $i = 1, \dots, \ell - 1$, we have

$$\lim_{\substack{k \to +\infty, \\ k \in K_1}} \frac{a_k^i}{a_k^\ell} = \lim_{\substack{k \to +\infty, \\ k \in K_1}} \frac{a_k^i}{a_k^{\hat{\imath}}} \frac{a_k^{\hat{\imath}}}{a_k^\ell} = 0,$$

and again the thesis is proved with $s = \ell$ and $K = K_1$.

Thus, the proof is concluded. □

Finally, in the next result, we establish the boundedness of the subsequence of Lagrange multipliers.

Theorem C.0.4. *In the conditions of Theorem 6.1.12, the subsequences* $\{\lambda_i(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^x}$, $i = 1, \dots, p$ *and* $\{\mu_j(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^x}$, $j = 1, \dots, q$, *defined as:*

$$\lambda_i(\boldsymbol{x}_k; \rho_k) = \begin{cases} \dfrac{\rho_k}{-g_i(\boldsymbol{x}_k)}, & \text{if } i \in \mathcal{G}_{log} \\ \nu \left(\dfrac{\max\{g_i(\boldsymbol{x}_k), 0\}}{\rho_k} \right)^{\nu - 1}, & \text{if } i \in \mathcal{G}_{ext} \end{cases}$$

$$\mu_j(\boldsymbol{x}_k; \rho_k) = \nu \left(\frac{|h_j(\boldsymbol{x}_k)|}{\rho_k} \right)^{\nu - 1}, \quad j = 1, \dots, q,$$

are bounded.

Proof. Recalling the expressions of $\lambda_i(\boldsymbol{x}; \rho)$, $i = 1, \dots, p$, and of $\mu_j(\boldsymbol{x}; \rho)$, $j = 1, \dots, q$, we can rewrite inequality (6.18) as

$$\begin{aligned} \Bigg(\nabla f(\boldsymbol{y}_k^l) + \sum_{i=1}^{p} \lambda_i(\boldsymbol{y}_k^l; \rho_k) \nabla g_i(\boldsymbol{y}_k^l) + \\ \sum_{j=1}^{q} \mu_j(\boldsymbol{y}_k^l; \rho) \nabla h_j(\boldsymbol{y}_k^l) \Bigg)^\top \boldsymbol{d}_k^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l}, \quad \forall\, l \in J_k \text{ and } k \in K_\rho^x, \end{aligned} \tag{C.6}$$

where $\boldsymbol{y}_k^l = \boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l$, with $t_k^l \in (0, 1)$ and $\hat{\alpha}_k^l \leq \alpha_k$.

We start by establishing that

$$\lim_{\substack{k \to +\infty \\ k \in K_\rho^x}} |\lambda_i(\boldsymbol{x}_k; \rho_k) - \lambda_i(\boldsymbol{y}_k^l; \rho_k)| = 0, \quad i \in \mathcal{G}_{\text{log}}, \quad \forall\, l \in J_k \tag{C.7}$$

In fact,

$$\begin{aligned} \left| \frac{\rho_k}{-g_i(\boldsymbol{x}_k)} - \frac{\rho_k}{-g_i(\boldsymbol{y}_k^l)} \right| &= \rho_k \left| \frac{g_i(\boldsymbol{x}_k) - g_i(\boldsymbol{y}_k^l)}{(-g_i(\boldsymbol{y}_k^l))(-g_i(\boldsymbol{x}_k))} \right| \\ &= \rho_k \frac{\left| \nabla g_i(\boldsymbol{u}_k^i)^\top \left(\boldsymbol{x}_k - \boldsymbol{y}_k^l \right) \right|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} \\ &\leq \rho_k \frac{\left\| \nabla g_i(\boldsymbol{u}_k^i) \right\| \left\| \boldsymbol{y}_k^l - \boldsymbol{x}_k \right\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|}, \end{aligned} \tag{C.8}$$

where $\boldsymbol{u}_k^i = \boldsymbol{x}_k + \tilde{t}_k^i(\boldsymbol{y}_k^l - \boldsymbol{x}_k)$ with $\tilde{t}_k^i \in (0,1)$.

Then, there is $c_1 > 0$ such that

$$\begin{aligned}\rho_k \left\|\nabla g_i(\boldsymbol{u}_k^i)\right\| \frac{\|\boldsymbol{y}_k^l - \boldsymbol{x}_k\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} &\leq \rho_k c_1 \frac{\|\boldsymbol{y}_k^l - \boldsymbol{x}_k\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} \\ &= \rho_k c_1 \frac{\|\boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l - \boldsymbol{x}_k\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} \\ &= \rho_k c_1 \frac{\|t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} \\ &= \rho_k c_1 \frac{t_k^l \hat{\alpha}_k^l \|\boldsymbol{d}_k^l\|}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|}.\end{aligned} \tag{C.9}$$

Now, we will prove that there is another constant $c_2 > 0$ such that

$$\frac{1}{|g_i(\boldsymbol{y}_k^l)|} \leq c_2 \frac{1}{|g_i(\boldsymbol{x}_k)|}. \tag{C.10}$$

Suppose, in order to arrive to a contradiction, that c_2 does not exist. This would imply that there exists $K_\rho^{\boldsymbol{x},g} \subseteq K_\rho^{\boldsymbol{x}} \subseteq K_\rho$ such that

$$\lim_{\substack{k\to+\infty \\ k\in K_\rho^{\boldsymbol{x},g}}} \frac{\frac{1}{|g_i(\boldsymbol{y}_k^l)|}}{\frac{1}{|g_i(\boldsymbol{x}_k)|}} = \lim_{\substack{k\to+\infty \\ k\in K_\rho^{\boldsymbol{x},g}}} \frac{|g_i(\boldsymbol{x}_k)|}{|g_i(\boldsymbol{y}_k^l)|} = +\infty. \tag{C.11}$$

Let us consider the case where

$$\lim_{\substack{k\to+\infty \\ k\in K_\rho^{\boldsymbol{x},g}}} |g_i(\boldsymbol{x}_k)| = 0.$$

Since $g_i(\boldsymbol{x}_k) < 0$ and $g_i(\boldsymbol{y}_k^l) < 0$ for all $k \in K_\rho^{\boldsymbol{x},g}$, by (C.11) there exists $\overline{k} \in \mathbb{N}$ such that, for all $k \geq \overline{k}$, $k \in K_\rho^{\boldsymbol{x},g}$, we have

$$-g_i(\boldsymbol{x}_k) > -g_i(\boldsymbol{y}_k^l) = -g_i(\boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l).$$

Using the Lipschitz continuity of g_i, $i = 1, \ldots, p$ and the fact that $\|\boldsymbol{d}_k^l\| = 1$ for all $l \in J_k$, we get

$$-g_i(\boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l) \geq -g_i(\boldsymbol{x}_k) - L_{g_i}\|t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l\| = -g_i(\boldsymbol{x}_k) - L_{g_i} t_k^l \hat{\alpha}_k^l.$$

The definition of K_ρ guarantees that

$$\alpha_{k+1} \leq \min\{\rho_k^\beta, (g_{\min})_k^2\}, \qquad \alpha_{k+1} = \theta_\alpha \alpha_k,$$

so that

$$\alpha_k \leq \frac{\min\{\rho_k^\beta, (g_{\min})_k^2\}}{\theta_\alpha} \tag{C.12}$$

Hence, since $\hat{\alpha}_k^l \leq \alpha_k$, we have

$$-g_i(\boldsymbol{x}_k) - L_{g_i} t_k^l \hat{\alpha}_k^l \geq -g_i(\boldsymbol{x}_k) - L_{g_i} t_k^l \frac{1}{\theta_\alpha}(g_i(\boldsymbol{x}_k))^2, \quad \forall k \geq \overline{k},\ k \in K_\rho^{\boldsymbol{x},g}$$

Thus,

$$\begin{aligned}\lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} \frac{-g_i(\boldsymbol{x}_k)}{-g_i(\boldsymbol{y}_k^l)} &= \lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} \frac{-g_i(\boldsymbol{x}_k)}{-g_i(\boldsymbol{x}_k+t_k^l\hat{\alpha}_k^l\boldsymbol{d}_k^l)}\\ &\leq \lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} \frac{-g_i(\boldsymbol{x}_k)}{-g_i(\boldsymbol{x}_k)-L_{g_i}t_k^l\frac{1}{\theta_\alpha}(g_i(\boldsymbol{x}_k))^2}=1,\end{aligned}$$

which leads to a contradiction, proving (C.10).

Now, by considering the other case

$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} |g_i(\boldsymbol{x}_k)| = c < +\infty,$$

we have

$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} \frac{-g_i(\boldsymbol{x}_k)}{-g_i(\boldsymbol{y}_k^l)} = \lim_{\substack{k\to+\infty\\ k\in K_\rho^{x,g}}} \frac{-g_i(\boldsymbol{x}_k)}{-g_i(\boldsymbol{x}_k+t_k^l\hat{\alpha}_k^l\boldsymbol{d}_k^l)} < +\infty.$$

Again, this leads to a contradiction, proving (C.10).

Hence, the existence of the constant $c_2 > 0$, (C.10), and recalling that $\hat{\alpha}_k^l \leq \alpha_k$, allow us to write

$$\frac{\rho_k c_1 t_k^l \hat{\alpha}_k^l}{|g_i(\boldsymbol{y}_k^l)||g_i(\boldsymbol{x}_k)|} \leq \frac{\rho_k c_1 c_2 t_k^l \alpha_k}{|g_i(\boldsymbol{x}_k)|^2}.$$

The instructions of Step 3 imply that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, so that $(g_{\min})_k = \min_{i\in\mathcal{G}_{\texttt{log}}}\{|g_i(\boldsymbol{x}_{k+1})|\} = \min_{i\in\mathcal{G}_{\texttt{log}}}\{|g_i(\boldsymbol{x}_k)|\}$. Recalling (C.12), we get

$$\frac{\rho_k\alpha_k}{(g_{\min})_k^2} \leq \frac{\rho_k}{\theta_\alpha}.$$

Then, recalling Theorem 6.1.10, (C.7) is proved.

Furthermore,

$$\lim_{\substack{k\to+\infty\\ k\in K_\rho^{x}}} |\lambda_i(\boldsymbol{x}_k;\rho_k)-\lambda_i(\boldsymbol{y}_k^l;\rho_k)| = 0, \quad i\in\mathcal{G}_{\texttt{ext}}, \quad \forall\, l\in J_k \tag{C.13}$$

can be established. In fact,

$$
\begin{aligned}
\left| \frac{\nu}{\rho_k^{\nu-1}} \left(\max\{g_i(\boldsymbol{x}_k), 0\}\right)^{\nu-1} \right. &- \left. \frac{\nu}{\rho_k^{\nu-1}} \left(\max\{g_i(\boldsymbol{y}_k^l), 0\}\right)^{\nu-1} \right| = \\
&= \frac{\nu}{\rho_k^{\nu-1}} \left| \max\{g_i(\boldsymbol{x}_k), 0\}^{\nu-1} - \max\{g_i(\boldsymbol{x}_k) + \nabla g_i(\boldsymbol{u}_k^i)^\top (\boldsymbol{x}_k - \boldsymbol{y}_k^l), 0\}^{\nu-1} \right| \\
\text{(Lemma C.0.2 – (C.2))} \quad &\leq \frac{\nu}{\rho_k^{\nu-1}} \left| g_i(\boldsymbol{x}_k) - g_i(\boldsymbol{x}_k) - \nabla g_i(\boldsymbol{u}_k^i)^\top (\boldsymbol{x}_k - \boldsymbol{y}_k^l) \right|^{\nu-1} \\
&= \frac{\nu}{\rho_k^{\nu-1}} |\nabla g_i(\boldsymbol{u}_k^i)^\top (\boldsymbol{x}_k - \boldsymbol{y}_k^l)|^{\nu-1} \leq \frac{\nu}{\rho_k^{\nu-1}} \left\| \nabla g_i(\boldsymbol{u}_k^i) \right\|^{\nu-1} \left\| \boldsymbol{x}_k - \boldsymbol{y}_k^l \right\|^{\nu-1} \\
&\leq c_3 \frac{\nu}{\rho_k^{\nu-1}} \left\| \left(\boldsymbol{x}_k - (\boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l) \right) \right\|^{\nu-1} = c_3 \frac{\nu}{\rho_k^{\nu-1}} (t_k^l \hat{\alpha}_k^l)^{\nu-1} \left\| \boldsymbol{d}_k^l \right\|^{\nu-1} \\
&\leq c_3 \frac{\nu}{\rho_k^{\nu-1}} (\hat{\alpha}_k^l)^{\nu-1} \left\| \boldsymbol{d}_k^l \right\|^{\nu-1} \leq c_3 \nu \left(\frac{\alpha_k}{\rho_k} \right)^{\nu-1} \left\| \boldsymbol{d}_k^l \right\|^{\nu-1} \\
&\leq c_3 \nu \left(\frac{\rho_k^{\beta-1}}{\theta_\alpha} \right)^{\nu-1} \left\| \boldsymbol{d}_k^l \right\|^{\nu-1} = c_3 \nu \theta_\alpha^{1-\nu} \rho_k^{(\beta-1)(\nu-1)} \left\| \boldsymbol{d}_k^l \right\|^{\nu-1},
\end{aligned}
$$

where $\boldsymbol{u}_k^i = \boldsymbol{x}_k + \tilde{t}_k^i (\boldsymbol{y}_k^l - \boldsymbol{x}_k)$ with $\tilde{t}_k^i \in (0,1)$, and $c_3 > 0$. Thus, using $\beta > 1$, $\nu \in (1,2]$, $\| \boldsymbol{d}_k^l \| = 1$ for all $l \in J_k$, and recalling Theorem 6.1.10, (C.13) is proved. Therefore, we have that

$$
\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\mathcal{X}}}} |\lambda_i(\boldsymbol{x}_k; \rho_k) - \lambda_i(\boldsymbol{y}_k^l; \rho_k)| = 0, \quad i = 1, \ldots, p, \quad \forall\, l \in J_k \tag{C.14}
$$

Let us now prove that

$$
\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\mathcal{X}}}} |\mu_j(\boldsymbol{x}_k; \rho_k) - \mu_j(\boldsymbol{y}_k^l; \rho_k)| = 0, \quad j = 1, \ldots, q, \quad \forall l \in J_k \tag{C.15}
$$

In fact, recalling that $\nu \in (1,2]$ so that $\nu - 1 \in (0,1]$, we have

$$
\begin{aligned}
\left| \nu \left| \frac{h_j(\boldsymbol{x}_k)}{\rho_k} \right|^{\nu-1} - \nu \left| \frac{h_j(\boldsymbol{y}_k^l)}{\rho_k} \right|^{\nu-1} \right| &= \frac{\nu}{\rho_k^{\nu-1}} \left| \left| h_j(\boldsymbol{x}_k) \right|^{\nu-1} - \left| h_j(\boldsymbol{x}_k) + \nabla h_j(\boldsymbol{u}_k^j)^\top (\boldsymbol{y}_k^l - \boldsymbol{x}_k) \right|^{\nu-1} \right| \\
\text{(Lemma C.0.2 – (C.1))} \quad &\leq \frac{\nu}{\rho_k^{\nu-1}} \left| h_j(\boldsymbol{x}_k) - h_j(\boldsymbol{x}_k) - \nabla h_j(\boldsymbol{u}_k^j)^\top (\boldsymbol{y}_k^l - \boldsymbol{x}_k) \right|^{\nu-1} \\
&= \frac{\nu}{\rho_k^{\nu-1}} \left| \nabla h_j(\boldsymbol{u}_k^j)^\top (\boldsymbol{y}_k^l - \boldsymbol{x}_k) \right|^{\nu-1} \\
&\leq \frac{\nu}{\rho_k^{\nu-1}} \left\| \nabla h_j(\boldsymbol{u}_k^j) \right\|^{\nu-1} \left\| (\boldsymbol{y}_k^l - \boldsymbol{x}_k) \right\|^{\nu-1},
\end{aligned} \tag{C.16}
$$

where $\boldsymbol{u}_k^j = \boldsymbol{x}_k + \tilde{t}_k^j (\boldsymbol{y}_k^l - \boldsymbol{x}_k)$, with $\tilde{t}_k^j \in (0,1)$.

Now, recalling that h_j, $j = 1, \ldots, q$ are continuously differentiable functions and $\boldsymbol{y}_k^l = \boldsymbol{x}_k + t_k^l \hat{\alpha}_k^l \boldsymbol{d}_k^l$, with $t_k^l \in (0,1)$ and $\| \boldsymbol{d}_k^l \| = 1$, from (C.16) and the fact that $\hat{\alpha}_k^l \leq \alpha_k$ we

can write, with $c_4 > 0$,

$$\begin{aligned}\left|\nu\left|\frac{h_j(\boldsymbol{x}_k)}{\rho_k}\right|^{\nu-1} - \nu\left|\frac{h_j(\boldsymbol{y}_k^l)}{\rho_k}\right|^{\nu-1}\right| &\leq \frac{\nu}{\rho_k^{\nu-1}} c_4 \left(t_k^l \hat{\alpha}_k^l\right)^{\nu-1} \\ &\leq c_4 \frac{\nu}{\rho_k^{\nu-1}} \left(t_k^l \alpha_k\right)^{\nu-1} \leq c_4 \nu \left(\frac{\alpha_k}{\rho_k}\right)^{\nu-1} \\ &\leq c_4 \nu \left(\frac{\rho_k^{\beta-1}}{\theta_\alpha}\right)^{\nu-1} = c_4 \nu \theta_\alpha^{1-\nu} \rho_k^{(\beta-1)(\nu-1)}.\end{aligned}$$

Given that $\beta > 1$, $\nu \in (1, 2]$, and recalling Theorem 6.1.10, we can conclude that (C.15) holds.

Now, we are able to prove the boundness of the sequences $\{\lambda_i(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^x}$, $i = 1, \ldots, p$ and $\{\mu_j(\boldsymbol{x}_k; \rho_k)\}_{k \in K_\rho^x}$, $j = 1, \ldots, q$.

In fact, we can rewrite (C.6) as

$$\begin{aligned}&\Bigg(\nabla f(\boldsymbol{y}_k^l) + \sum_{i=1}^{p} \nabla g_i(\boldsymbol{y}_k^l)\lambda_i(\boldsymbol{x}_k; \rho_k) + \\ &+ \sum_{i=1}^{p} \nabla g_i(\boldsymbol{y}_k^l)(\lambda_i(\boldsymbol{y}_k^l; \rho_k,) - \lambda_i(\boldsymbol{x}_k; \rho_k)) + \sum_{j=1}^{q} \nabla h_j(\boldsymbol{y}_k^l)\mu_j(\boldsymbol{x}_k; \rho_k) + \\ &+ \sum_{j=1}^{q} \nabla h_j(\boldsymbol{y}_k^l)(\mu_j(\boldsymbol{y}_k^l; \rho_k) - \mu_j(\boldsymbol{x}_k; \rho_k))\Bigg)^\top \boldsymbol{d}_k^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l}, \quad \forall l \in J_k \text{ and } k \in K_\rho^{\boldsymbol{x}}.\end{aligned} \tag{C.17}$$

Let

$$\begin{aligned}\{a_k^1, \ldots, a_k^p\} &= \{\lambda_1(\boldsymbol{x}_k; \rho_k), \ldots, \lambda_p(\boldsymbol{x}_k; \rho_k)\}, \\ \{a_k^{p+1}, \ldots, a_k^{p+q}\} &= \{\mu_1(\boldsymbol{x}_k; \rho_k), \ldots, \mu_q(\boldsymbol{x}_k; \rho_k)\}.\end{aligned}$$

Assume, by contradiction, that there exists at least one index $\ell \in \{1, \ldots, p + q\}$ such that

$$\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x}}}} |a_k^\ell| = +\infty. \tag{C.18}$$

Hence, the sequence $\{a_k^i\}$, $i = 1, \ldots, p + q$, cannot be all convergent to zero. Then, from Lemma C.0.3, there exists an infinite subset $K_\rho^{\boldsymbol{x},\boldsymbol{a}} \subseteq K_\rho^{\boldsymbol{x}}$ and an index $s \in \{1, \ldots, p + q\}$ such that,

$$\lim_{\substack{k \to +\infty \\ k \in K_\rho^{\boldsymbol{x},\boldsymbol{a}}}} \frac{a_k^i}{|a_k^s|} = z_i, \quad |z_i| < +\infty, \quad i = 1, \ldots, p + q \tag{C.19}$$

If there is an unique index ℓ that satisfies (C.18), then $s = \ell$. If we have more than one index satisfying the equation, then s is selected as one of the indexes such that $\{a_k^s\}_{k \in K_\rho^{\boldsymbol{x},\boldsymbol{a}}}$ tends to $+\infty$ faster than the others. Note also that

$$z_s = 1, \qquad \text{and} \qquad |a_k^s| \to +\infty. \tag{C.20}$$

Dividing the relation (C.17) by $|a_k^s|$, we have

$$\left(\frac{\nabla f(\boldsymbol{y}_k^l)}{|a_k^s|} + \sum_{i=1}^{p} \frac{\nabla g_i(\boldsymbol{y}_k^l) a_k^i}{|a_k^s|} + \sum_{i=1}^{p} \nabla g_i(\boldsymbol{y}_k^l) \frac{\lambda_i(\boldsymbol{y}_k^l;\rho_k) - \lambda_i(\boldsymbol{x}_k;\rho_k)}{|a_k^s|} + \sum_{j=1}^{q} \frac{\nabla h_j(\boldsymbol{y}_k^l) a_k^{p+j}}{|a_k^s|} + \sum_{j=1}^{q} \nabla h_j(\boldsymbol{y}_k^l) \frac{\mu_j(\boldsymbol{y}_k^l;\rho_k) - \mu_j(\boldsymbol{x}_k;\rho_k)}{|a_k^s|}\right)^\top \boldsymbol{d}_k^l \geq -\frac{\xi(\hat{\alpha}_k^l)}{\hat{\alpha}_k^l |a_k^s|}, \quad \forall l \in J_k \text{ and } k \in K_\rho^{\boldsymbol{x},\boldsymbol{a}}. \tag{C.21}$$

Reasoning as for (6.18), by Assumption 6.1.5 we can extract a further subsequence $K_\rho^{\boldsymbol{x},\boldsymbol{a},\boldsymbol{D}} \subseteq K_\rho^{\boldsymbol{x},\boldsymbol{a}}$ such that for all $k \in K_\rho^{\boldsymbol{x},\boldsymbol{a},\boldsymbol{D}}$ it holds $J_k = \bar{J}$, $\boldsymbol{d}_k^l = \bar{\boldsymbol{d}}^l$ for all $l \in \bar{J}$, and $\boldsymbol{D}_k = \overline{\boldsymbol{D}} = \{\bar{\boldsymbol{d}}^l\}_{l\in\bar{J}}$. Since $\lim_{k\in K_\rho^{\boldsymbol{x},\boldsymbol{a},\boldsymbol{D}}} \boldsymbol{x}_k = \overline{\boldsymbol{x}}$, by Assumption 6.1.5 and Proposition 6.1.4, and using $\varepsilon \in (0, \min\{\bar{\varepsilon}, \varepsilon^*\}]$, for sufficiently large $k \in K_\rho^{\boldsymbol{x},\boldsymbol{a},\boldsymbol{D}}$ we have $T_X(\overline{\boldsymbol{x}}) = T_X(\boldsymbol{x}_k, \varepsilon) = \text{cone}(\boldsymbol{D}_k \cap T_X(\boldsymbol{x}_k, \varepsilon)) = \text{cone}(\overline{\boldsymbol{D}})$.

Taking the limit for $k \to +\infty$ and $k \in K_\rho^{\boldsymbol{x},\boldsymbol{a},\boldsymbol{D}}$, and using (C.14), (C.15), and (C.19), we obtain

$$\left(\sum_{i=1}^{p} z_i \nabla g_i(\overline{\boldsymbol{x}}) + \sum_{j=1}^{q} z_{p+j} \nabla h_j(\overline{\boldsymbol{x}})\right)^\top \bar{\boldsymbol{d}}^l \geq 0, \quad \forall \bar{\boldsymbol{d}}^l \in \overline{\boldsymbol{D}}. \tag{C.22}$$

We recall that $\overline{\boldsymbol{x}}$ satisfies the MFCQ conditions. Let $\boldsymbol{d}$ be the direction satisfying condition (b) of Definition 2.1.22. For every $\boldsymbol{d} \in T_X(\overline{\boldsymbol{x}})$, there exist nonnegative numbers β_l such that

$$\boldsymbol{d} = \sum_{\bar{\boldsymbol{d}}^l \in \overline{\boldsymbol{D}}} \beta_l \bar{\boldsymbol{d}}^l. \tag{C.23}$$

Thus, from (C.22) and (C.23), we obtain

$$\begin{aligned}
&\left(\sum_{i=1}^{p} z_i \nabla g_i(\overline{\boldsymbol{x}}) + \sum_{j=1}^{q} z_{p+j} \nabla h_j(\overline{\boldsymbol{x}})\right)^\top \boldsymbol{d} = \\
&= \sum_{\bar{\boldsymbol{d}}^l \in \overline{\boldsymbol{D}}} \beta_l \left(\sum_{i=1}^{p} z_i \nabla g_i(\overline{\boldsymbol{x}}) + \sum_{j=1}^{q} z_{p+j} \nabla h_j(\overline{\boldsymbol{x}})\right)^\top \bar{\boldsymbol{d}}^l \\
&= \sum_{\bar{\boldsymbol{d}}^l \in \overline{\boldsymbol{D}}} \beta_l \sum_{i=1}^{p} z_i \nabla g_i(\overline{\boldsymbol{x}})^\top \bar{\boldsymbol{d}}^l + \sum_{\bar{\boldsymbol{d}}^l \in \overline{\boldsymbol{D}}} \beta_l \sum_{j=1}^{q} z_{p+j} \nabla h_j(\overline{\boldsymbol{x}})^\top \bar{\boldsymbol{d}}^l \geq 0.
\end{aligned} \tag{C.24}$$

Considering Definition 2.1.22, from (C.24) it follows

$$\sum_{i=1}^{p} z_i \nabla g_i(\overline{\boldsymbol{x}})^\top \boldsymbol{d} \geq 0. \tag{C.25}$$

Theorem 6.1.10 and the definition of z_i for $i \in \{1, \ldots, p\}$, guarantee

$$z_i = 0, \quad \text{for all} \quad i \notin I_+(\overline{\boldsymbol{x}}). \tag{C.26}$$

Since $\overline{x}$ satisfies the MFCQ conditions, (C.25) implies

$$z_i = 0, \quad \text{for all } i \in I_+(\overline{x}). \tag{C.27}$$

Applying (C.26) and (C.27) to (C.22) we get

$$\left(\sum_{j=1}^{q} z_{p+j} \nabla h_j(\overline{x})\right)^\top \overline{d} \geq 0, \quad \text{for all} \quad \overline{d} \in \overline{D}, \tag{C.28}$$

using again Definition 2.1.22 and (C.28), we obtain

$$z_{p+j} = 0, \quad \text{for all} \quad j \in \{1, \ldots, q\}. \tag{C.29}$$

In conclusion, we get (C.26), (C.27), and (C.29), contradicting (C.20) and this concludes the proof. □