STATE UNIVERSITY OF MARINGÁ

CENTER OF EXACT SCIENCES

DEPARTAMENT OF MATHEMATICS

GRADUATE PROGRAM IN MATHEMATICS

(Doctorate)

ANDERSON ERVINO SCHWERTNER

Derivative-free Low Order-Value Optimization[1]

Otimização de Menor Valor Ordenado Sem Derivadas

Maringá - PR

2023

[1]This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

Anderson Ervino Schwertner

Derivative-free Low Order-Value Optimization

Otimização de Menor Valor Ordenado Sem Derivadas

Doctorate thesis submitted to the Graduate Program in Mathematics of the Department of Mathematics, Center of Exact Sciences of the State University of Maringá, as a requirement to obtain the title of Ph.D. in Mathematics.
Concentration area: Applied Mathematics.

Advisor: Dr. Francisco Nogueira Calmon Sobral

Maringá - PR
2023

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Setorial BSE-DMA-UEM, Maringá, PR, Brasil)

Schwertner, Anderson Ervino
S415d Derivative-free low order-value optimization = Otimização de menor valor ordenado sem derivadas / Anderson Ervino Schwertner. -- Maringá, 2023.
xx 86 f. : il., color.

Orientador: Prof. Dr. Francisco Nogueira Calmon Sobral.
Tese (doutorado) - Universidade Estadual de Maringá, Centro de Ciências Exatas, Programa de Pós-Graduação em Matemática - Área de Concentração: Matemática Aplicada, 2023.

1. Otimização sem derivadas. 2. Otimização de menor valor ordenado. 3. Método de região de confiança. 4. Análise de complexidade do pior caso. 5. Derivative-free optimization. 6. Low order-value optimization. 7. Trust region method. 8. Worst case complexity analysis. I. Sobral, Francisco Nogueira Calmon, orient. II. Universidade Estadual de Maringá. Centro de Ciências Exatas. Programa de Pós-Graduação em Matemática - Área de Concentração: Matemática Aplicada. III. Título.

CDD 22.ed. 519.6
Edilson Damasio CRB9-1.123

ANDERSON ERVINO SCHWERTNER

DERIVATIVE-FREE LOW ORDER-VALUE OPTIMIZATION

Tese apresentada ao Programa de Pós-Graduação em Matemática do Departamento de Matemática, Centro de Ciências Exatas da Universidade Estadual de Maringá, como parte dos requisitos necessários para a obtenção do título de Doutor em Matemática tendo a Comissão Julgadora composta pelos membros:

Aprovado em: 23 de fevereiro de 2023.
Local de defesa: Videoconferência – Google Meet (https://meet.google.com/uzw-bnxg-mhd)

I dedicate this thesis to all children who dream
of building a better future through education.

Acknowledgments

This endeavor would not have been possible without the dedication, patience, companionship, and essential support of my supervisor. I am also grateful for your great lessons, for believing in me, and for showing me that we can do much more when we have focus and the correct guidance.

I would like to extend my sincere thanks to the professors, employees, and colleagues of the Graduate Program in Mathematics, and all those who contributed to the conclusion of this journey. In particular, I am grateful for the countless opportunities we had to be together, discussing Mathematics or talking about things in life.

I dedicate a special thanks to the members of the defense committee for accepting the invitation and dedicating their time to read and revise this work.

I am also thankful to IMPA and OBMEP for showing me that Mathematics can be very funny and that science is a fascinating experience.

I would like to mention my appreciation and gratitude to God for my family. I am also grateful for His grace, care, and love for my family and me. I thank my beloved wife, for her patience and affection, for always being by my side and encouraging me to keep moving forward. I am grateful to my parents for being constant examples and believing in my dreams. I am thankful to my brother and sister-in-law for supporting me and being examples of perseverance.

Lastly, I would like to thank the financial support that I received from CAPES, CNPq, and Fundação Araucária during my academic journey.

“Do your best, in the condition you have,

while you don’t have better conditions, to do even better.”

Mario Sergio Cortella

Abstract

The development of derivative-free optimization (DFO) methods was driven by the growing need to solve complex and diverse problems, particularly, problems for which the derivatives of the objective function and constraints are not available. This is the case for many practical applications in science, medicine, and engineering, among others. For example, in situations where the objective function is of black-box type, it has no analytical expression or it has a high computational cost. There are several different approaches to DFO methods, among which we can mention direct and pattern search, model-based, augmented lagrangian, among others. Despite the wide variety of methods found in the literature, we are unaware of methods dedicated to solving low order-value optimization (LOVO) problems, in which we seek to minimize the minimum among a finite number of function values within a feasible set. In this work, we are interested in the constrained nonlinear optimization LOVO problem, whose feasible set is convex, closed, and nonempty, and each component function is black-box and continuously differentiable. We also assume that it is simple to compute the orthogonal projection of an arbitrary point onto the feasible set. We developed a derivative-free trust-region algorithm for constrained LOVO problems with convergence to weakly critical points. Under suitable conditions, we establish global convergence and worst case complexity results. We discuss the construction of linear and quadratic models suitable for derivative-free optimization and extend the concept to underdetermined quadratic models based on approximate values of the function. Finally, we present an efficient implementation of our algorithm, as well as numerical results.

Keywords: Derivative-free Optimization, Low Order-Value Optimization, Trust-Region Method, Worst Case Complexity Analysis.

Resumo

O desenvolvimento dos métodos de otimização sem derivadas (DFO) foi impulsionado pela necessidade crescente de resolver problemas complexos e diversos, em especial, problemas para os quais as derivadas da função objetivo e das restrições não estão disponíveis. Este é o caso de diversas aplicações práticas encontradas na ciência, medicina e engenharia, entre outros. Por exemplo, em situações onde a função objetivo é do tipo caixa preta (*black-box*), não possui expressão analítica, ou ainda, possui um alto custo computacional. Existem diversas abordagens distintas para DFO, dentre as quais podemos citar busca direta, busca padrão, baseada em modelos, lagrangianos aumentados, entre outros. Apesar da grande variedade de métodos encontrados na literatura, desconhecemos métodos dedicados a solução de problemas de otimização de menor valor ordenado (LOVO), nos quais buscamos minimizar o mínimo entre um número finito de valores de função em um conjunto viável. Neste trabalho estamos interessados no problema de otimização LOVO não linear restrito, cujo conjunto viável é convexo, fechado e não vazio, e cada função componente é do tipo *black-box* e continuamente diferenciável. Também assumimos que é simples calcular a projeção ortogonal de um ponto arbitrário sobre o conjunto viável. Desenvolvemos um algoritmo de região de confiança sem derivadas para problemas LOVO restritos com convergência para pontos fracamente críticos. Sob condições adequadas, estabelecemos resultados de convergência global e de complexidade do pior caso. Discutimos a construção de modelos lineares e quadráticos adequados para otimização sem derivadas e estendemos o conceito para modelos quadráticos subdeterminados baseados em valores aproximados da função. Por fim, apresentamos uma implementação eficiente de nosso algoritmo, bem como resultados numéricos promissores.

Palavras-chave: Otimização sem Derivadas, Otimização de Menor Valor Ordenado, Método de Região de Confiança, Análise de Complexidade do Pior Caso.

List of Figures

List of Tables

List of Symbols and Notations

$\mathcal{I}$	Set of indexes.
$\mathcal{Y}$	Set of sample points (or interpolation points).
$\lvert\mathcal{Y}\rvert$	Cardinality of set $\mathcal{Y}$.
$B(x,\delta)$	Open ball centered at x with radius δ.
$\overline{B}(x,\delta)$	Closed ball centered at x with radius δ.
$\mathcal{P}_n^a(\mathbb{R})$	Space of polynomials of degree at most a in n variables with real coefficients.
$dim(\mathcal{P}_n^a(\mathbb{R}))$	Dimension of space $\mathcal{P}_n^a(\mathbb{R})$.
$\lvert\alpha\rvert$	Absolute value of α.
$\lVert x\rVert$	Euclidean norm of vector x.
$\lVert x\rVert_\infty$	Maximum norm of vector x.
$\lVert\mathbf{A}\rVert$	Euclidean induced norm of matrix $\mathbf{A}$.
$\mathbf{A}^{-1}$	Inverse matrix of $\mathbf{A}$.
$\mathbf{A}^\dagger$	Pseudoinverse matrix of $\mathbf{A}$.
$\mathbf{I}_n$	Identity matrix of order n.
$\mathcal{O}(n)$	Big O notation for linear behavior.
$\mathcal{O}(n^2)$	Big O notation for quadratic behavior.

Contents

Chapter 1

Introduction

There are several optimization problems for which information regarding the first and second-order derivatives is unavailable, unreliable, or demands a high computational cost. This is the case for many scientific, social, medical, engineering, and, more recently, artificial intelligence applications [5, 24, 27, 45]. In this sense, derivative-free optimization algorithms are convenient tools to handle such problems since, essentially, they employ only objective function values. Detailed surveys of derivative-free optimization methods can be found in [45, 63], and theoretical aspects can be revisited in [5, 27].

The development of these methods goes back to the work of Nelder and Mead [54], Spendley et al. [68], and Winfield [76] in the 1960s. After a period of little activity, the last 25 years have seen a resurgence and an increase in the effort dedicated to developing efficient methods for derivative-free optimization [13], and they became popular due to the good numerical performance of the algorithms proposed by Powell [57, 58, 59]. Among the various classes of existing derivative-free optimization methods, we will focus on model-based methods. In particular, we are interested in the derivative-free trust-region method. In this framework, we replace the information about the derivative of the objective function with the derivative of a model, which is generally smooth, easy to evaluate, and accurate in a neighborhood of the point of interest [63]. There are several derivative-free trust-region algorithms described in the literature, which can be applied to unconstrained problems [5, 19, 23, 27, 28, 32, 33, 57, 58, 60, 65, 75], bound constrained [4, 38, 59, 73], linearly constrained [40, 61], convex constrained [17, 42, 72], and general constrained [12, 18, 24, 34, 71] problems, as well as composite nonsmooth optimization [35, 37, 46], and multi-objective optimization [7, 64, 69] problems, among others.

In this work, we are interested in Low Order-Value Optimization problems (LOVO) [2]. The LOVO problem involves minimizing the minimum among a finite number of function values within a feasible set, and has several practical applications such as protein alignment [1, 2, 48, 49], robust parameter estimation [2, 15, 31, 66], portfolio optimization [10, 48], hidden pattern search [2], computer vision [47], fitting Nash-Equilibrium models [47], robust response surface methodology [56], among others. More specifically, we are interested in the following constrained LOVO problem:

$$\min_{x\in\Omega} f_{min}(x) = \min_{x\in\Omega} \min\{f_1(x), \dots, f_r(x)\}, \tag{1.1}$$

where $\Omega \subset \mathbb{R}^n$ is a nonempty closed convex set and $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, r$, are continuously differentiable black-box functions. Note that function f_{min} is not necessarily smooth, and for points in its domain where the derivative is well-defined, it simply cannot be accessed because each component function f_i is of black-box type. In order to solve problem (1.1), we developed a derivative-free trust-region algorithm specialized in LOVO black-box problems.

In recent years, several authors have developed LOVO versions of classical continuous optimization methods, such as the Levenberg-Marquardt [15, 31, 66], Augmented Lagrangian [2, 47], Trust-Region [1], Line-Search [2, 49] and Coordinate-Search [48] algorithms. However, to the best of our knowledge, the method we propose is the first derivative-free algorithm specially designed for LOVO black-box problems.

In Chapter 2, we organize several results in the literature about bounds that appear in derivative-free trust-region algorithms based on linear and quadratic models. When dealing with model-based methods, it is natural to ask ourselves how to measure and guarantee the quality of the model to be built during the iterations of the algorithm. An indicator of the quality of the model is the degree to which the model approximates the objective function f and its derivatives in a neighborhood of interest, which usually also involves some assumptions about the smoothness of the model and the function [27, 45, 72]. So that we can guarantee the quality of the model, the set of interpolation points must have good geometric properties, which typically involves ensuring that the system of equations that defines the interpolation model has a solution and that its norm is bounded [27, 72]. All these characteristics are related to the bounds presented in Chapter 2, and these constants are given explicitly by the quality of the sample set, the dimensions of the problem and polynomial space, and the number of sample points. We consider linear and quadratic determined models and underdetermined quadratic models, particularly minimum Frobenius norm models. We extend some results to allow “inexact” interpolation sets, where some numerical errors are expected in the evaluation of the objective function or in the construction of the models. This new approach is related to linear and quadratic models build from support vector regression strategies [72]. We also provide clearer proofs than those already existing in the literature for the underdetermined case.

The main contribution of this work resides in Chapter 3, where we propose the first derivative-free trust-region algorithm designed for convex-constrained LOVO problems. Our algorithm is mainly based on the algorithms proposed in [17, 72], and on the ideas discussed by [1]. Since we do not specify the construction of the models, we allow a certain freedom in their choice as long as they satisfy a quality condition. As in the algorithm proposed by [72], we employ two distinct radii, one for the trust-region and the other for the sample region, which meets the theoretical need to allow the term that controls the quality of the model to converge to zero, while practical experiments show us that it is desirable that the radius of the trust-region to be as large as possible. Under assumptions common to other model-based derivative-free optimization algorithms, we also present global convergence results for weakly critical points, as well as an interpretation of these results under the classic theory of LOVO problems. Finally, we perform the worst-case analysis and show that the algorithm has the expected behavior for model-based methods.

Chapter 4 is dedicated to exposing the implementation details of our algorithm, the description of the numerical tests performed, as well as the results we found. In this sense, the theory of global convergence and worst-case analysis, established in Chapter 3, is of great importance as they provide us with indications about the correct implementation of the algorithm’s mechanisms. Our implementation is also compared with other algorithms capable of solving piecewise continuous derivative-free optimization problems, although they do not have the ability to explore the properties of LOVO problem. The tests are carried out choosing three sets of test problems, one unrestricted and two with box constraints.

Finally, in Chapter 5, we present the conclusions of this work and several suggestions for future research, involving issues related to models for optimization without derivatives, as well as extensions of the theory developed for our algorithm. We also suggest several practical enhancements that can be implemented to improve its performance, making it more efficient and robust.

Chapter 2

Polynomial models and complexity constants

In this chapter, we study polynomial models, commonly used in derivative-free trust-region algorithms. In particular, we discuss the conditions under which they provide good approximations to the true function. This chapter is organized as follows. In Section 2.1 we present a brief introduction about models for derivative-free optimization, and we point out the importance of establishing accurate complexity constants to understand the numerical behavior of optimization algorithms. Section 2.2 is concerned with bounds to linear and quadratic polynomial models which are uniquely determined by the sample set. Section 2.3 is the main contribution of this chapter and discusses error bounds for underdetermined quadratic models, since they are the most efficient models in practice [58]. Finally, all complexity constants that appear in this chapter are organized and summarized in Section 2.4.

The text of this chapter is an extended version of the results presented in [67], which are currently under review.

2.1 Short overview

Polynomial interpolation is one of the main aspects of model-based derivative-free algorithms. Given a function $f : \mathbb{R}^n \to \mathbb{R}$ and a set of interpolation points $\mathcal{Y} = \{y^0, \ldots, y^p\}$, the interpolating polynomial $\mathrm{m} \in \mathcal{P}_n^a(\mathbb{R})$ is such that

$$\mathrm{m}(y^i) = f(y^i), \quad i = 0, \ldots, p, \tag{2.1}$$

where $\mathcal{P}_n^a(\mathbb{R})$ is the space of polynomials of degree at most a in n variables. It is well understood that quadratic polynomials ($a = 2$) provide a good approximation, being able to capture the curvature of f using a reasonable amount of interpolation points [27, p.35]. In what follows, the set of interpolation points $\mathcal{Y}$ is also called the sample set, a well-established term in the literature on derivative-free optimization [5, 27, 45, 63].

More recently, the worst-case complexity analysis of optimization algorithms became popular. Although most of the bounds obtained are very pessimistic, the analysis provide further insights into the parameters and the dependence on the problems' dimensions.

When studying worst-case complexity of derivative-free trust-region algorithms using linear or quadratic polynomials, one has to deal with bounds related to the geometric quality of $\mathcal{Y}$ and to the Hessian of the model (in the quadratic case). In [27], several bounds were provided, which explicitly show the dependence on the problems' dimensions, that is, the dimension and the number of interpolation points. Such bounds were used, for example, in the excellent complexity analysis of [13, 35]. The complexity bounds generated the desire of performing derivative-free optimization on smaller subspaces, to further decrease the impact of dimensionality in the construction of models [14].

In [72], some error bounds were obtained for a relaxed version of condition (2.1)

$$|\mathrm{m}(y^i) - f(y^i)| \leq \kappa\delta^2, \quad i = 0, \ldots, p, \tag{2.2}$$

where κ is a constant and δ can be viewed as the precision of the model [61, 72] or just as the trust-region radius [27]. Condition (2.2) is related to linear and quadratic models build from support vector regression strategies. Unfortunately, only the determined case is analyzed in [72], that is, the case where $p = dim(\mathcal{P}_n^a(\mathbb{R})) - 1$. This work extends the bounds obtained in [72] to the underdetermined case, which is known to have better performance in practical implementations, and organize all the bounds associated with linear and quadratic polynomial models in literature, making clear their dependence on the dimension and quality of the interpolation set.

2.2 Determined models

Let us assume that $\mathcal{Y} \subset \overline{B}(y^0, \delta)$. Determined interpolation models need that the number of sample points in $\mathcal{Y}$ equals the dimension of $\mathcal{P}_n^a(\mathbb{R})$. In our notation, the number of sample points is given by $|\mathcal{Y}| = p$, and the dimension of the polynomial space is given by $dim(\mathcal{P}_n^a(\mathbb{R})) = q$. In the linear case, we assume that $p = n = q$ and define matrices

$$\mathbf{L}_L = \begin{bmatrix} (y^1 - y^0)^T \\ \vdots \\ (y^n - y^0)^T \end{bmatrix} = \begin{bmatrix} y_1^1 - y_1^0 & \cdots & y_n^1 - y_n^0 \\ \vdots & \ddots & \vdots \\ y_1^n - y_1^0 & \cdots & y_n^n - y_n^0 \end{bmatrix}, \text{ and } \hat{\mathbf{L}}_L = \frac{1}{\delta}\mathbf{L}_L. \tag{2.3}$$

In the quadratic case, we assume $p = (n^2 + 3n)/2 = q$ and define the matrix

$$\mathbf{L}_Q = \begin{bmatrix} \left(\overline{\varphi}(y^1 - y^0)\right)^T \\ \vdots \\ \left(\overline{\varphi}(y^q - y^0)\right)^T \end{bmatrix},$$

where $\overline{\varphi}(x) = \left[x_1, x_2, \ldots, x_n, \frac{1}{2}x_1^2, x_1x_2, x_1x_3, \ldots, x_1x_n, \frac{1}{2}x_2^2, \ldots, x_{n-1}x_n, \frac{1}{2}x_n^2\right]^T$ is the vector whose elements form the natural basis of monomials in $\mathcal{P}_n^2(\mathbb{R})$, as defined in [27, Section 3.1]. We also consider the matrix

$$\hat{\mathbf{L}}_Q = \mathbf{L}_Q \begin{bmatrix} \mathbf{D}_L^{-1} & 0 \\ 0 & \mathbf{D}_Q^{-1}, \end{bmatrix} \tag{2.4}$$

where $\mathbf{D}_L = \delta \mathbf{I}_n \in \mathbb{R}^{n \times n}$, and $\mathbf{D}_Q = \delta^2 \mathbf{I}_{(q-n)} \in \mathbb{R}^{(q-n)\times(q-n)}$. Matrices $\mathbf{L}_L$ and $\mathbf{L}_Q$ are related with the construction of determined interpolation models, while (2.3) and (2.4) are their respective scaled versions, mainly used for theoretical purposes.

We will assume that the following assumptions are valid.

Assumption 2.1. *∇f is Lipschitz continuous with constant L in a sufficiently large open bounded domain $\mathcal{X} \subset \mathbb{R}^n$.*

Assumption 2.2. *f is bounded below.*

Assumption 2.3. *If the model to be built is linear, then the matrix $\mathbf{L}_L$ is non-singular and there is a constant $\kappa_L > 0$ such that $\left\|\hat{\mathbf{L}}_L^{-1}\right\| \leq \kappa_L$. If the model is quadratic, then the matrix $\mathbf{L}_Q$ is nonsingular and there is a constant $\kappa_Q > 0$ such that $\left\|\hat{\mathbf{L}}_Q^{-1}\right\| \leq \kappa_Q$.*

The next assumption is related to the relaxed concept of interpolation models considered in this chapter, which follows condition (2.2).

Assumption 2.4. *There is a constant $\kappa \geq 0$ such that for every $y^i \in \mathcal{Y} \subset \overline{B}(y^0, \delta)$, $|\mathrm{m}(y^i) - f(y^i)| \leq \kappa\delta^2$.*

Note that Assumption 2.4 includes the creation of "exact" interpolation models and also models under uncertainty [72], under numerical errors or even when we can control the precision in which f is calculated [11]. The next theorem provides the error bounds for determined polynomial interpolation under Assumption 2.4.

Theorem 2.5. *Suppose that Assumptions 2.1, 2.2, 2.3 and 2.4 hold. Then, for all $x \in \mathcal{X} \cap \overline{B}(y^0, \delta)$, if the model is linear*

$$\begin{aligned} \left\|\nabla^2 \mathrm{m}(x)\right\| &= 0, \\ \|\nabla f(x) - \nabla \mathrm{m}(x)\| &\leq \left(L + \left(\frac{1}{2}L + 2\kappa\right)\kappa_L\sqrt{n}\right)\delta, \\ |f(x) - \mathrm{m}(x)| &\leq \left(\frac{1}{2}L + \kappa + \left(\frac{1}{2}L + 2\kappa\right)\kappa_L\sqrt{n}\right)\delta^2, \end{aligned}$$

and, if the model is quadratic,

$$\begin{aligned} \left\|\nabla^2 \mathrm{m}(x)\right\| &\leq 2\kappa_Q\sqrt{2q}(\kappa + L), \\ \|\nabla f(x) - \nabla \mathrm{m}(x)\| &\leq \left(2\kappa_Q\sqrt{q}\left(1 + \sqrt{2}\right)(\kappa + L)\right)\delta, \\ |f(x) - \mathrm{m}(x)| &\leq \left(\frac{1}{2}L + \kappa + \kappa_Q\sqrt{q}\left(2 + 3\sqrt{2}\right)(\kappa + L)\right)\delta^2. \end{aligned}$$

Proof. See [72, Theorem 2]. □

Observe that the error bounds obtained in Theorem 2.5 depend on the properties of the objective function, the sample set, and the relaxed concept of interpolation models expressed by Assumption 2.4. According to [72, p. 15], it is possible to obtain better bounds, with the powers of delta increased by one in both cases, similarly to [27, Theorem 3.16]. However, it is necessary to have stronger properties than Assumptions 2.3 and 2.4, such as assuming that the objective function f is twice continuously differentiable and $\nabla^2 f$ is Lipschitz continuous in a sufficiently large open bounded domain.

In order to ensure Assumption 2.3, we need to make additional hypotheses about the geometry of the sample set $\mathcal{Y}$. The existence and uniqueness of model m, also known as *poisedness*, is not enough. In linear interpolation, for example, this can be achieved by assuming that the points in $\mathcal{Y}$ are sufficiently affinely linear independent [73]. Here, we assume that the set $\mathcal{Y}$ is Λ-poised on the ball $\overline{B}(y^0, \delta)$, for some $\Lambda > 0$. This definition is taken from [27] and its reproduced below. This concept will be better discussed in Section 2.3 for underdetermined polynomials.

Definition 2.6. *[27, Definition 3.6] Let $\phi = \{\phi_0(x), \phi_1(x), \ldots, \phi_p(x)\}$ be a basis in $\mathcal{P}_n^a$ and $\Lambda > 0$. A set $\mathcal{Y} = \{y^0, \ldots, y^p\} \subset \mathcal{X}$ is Λ-poised in $\overline{B}(y^0, \delta)$ (in the interpolation sense) if for each $x \in \overline{B}(y^0, \delta)$ there exists $\lambda(x) \in \mathbb{R}^{p+1}$ such that*

$$\sum_{i=0}^{p} \lambda_i(x)\phi_j(y^i) = \phi_j(x),$$

for all $j = 0, \ldots, p$, with $\|\lambda(x)\|_\infty \leq \Lambda$.

Note that the definition of Λ-poisedness does not depend on the choice of basis [72, p. 16]. Furthermore, the constant Λ does not depend on the scale of the sample set [27, Lemma 3.8] and is invariant concerning coordinate shifts [27, Lemma 3.9]. The following lemma, adapted from [72], shows that Assumption 2.3 is valid when the models to be built are linear or quadratic and $\mathcal{Y}$ is Λ-poised.

Lemma 2.7. *Let $\Lambda > 0$. If $\mathcal{Y} = \{y^0, y^1, \ldots, y^n\}$ is Λ-poised in $\overline{B}(y^0, \delta)$ with respect to the basis ϕ of $\mathcal{P}_n^1$, then $\left\|\hat{\mathbf{L}}_L^{-1}\right\| \leq \Lambda\sqrt{n}$. If $\mathcal{Y} = \{y^0, y^1, \ldots, y^q\}$ is a Λ-poised set in $\overline{B}(y^0, \delta)$ with respect to the basis ϕ of $\mathcal{P}_n^2$, then, $\left\|\hat{\mathbf{L}}_Q^{-1}\right\| \leq 4\Lambda\sqrt{(q+1)^3}$.*

Proof. See [72, Lemma 8] and [72, Lemma 10]. □

2.3 Underdetermined models

The main drawback of determined polynomials is the high number of interpolation points needed in the quadratic case. One could use only linear polynomials, but quadratic polynomials are known to better explore the curvature of f. Therefore, we are interested in building quadratic polynomials using less than q interpolation points. Now we assume that $n < p < q$ and $q = dim(\mathcal{P}_n^2(\mathbb{R})) - 1 = (n^2 + 3n)/2$.

When dealing with classical interpolation theory, in the sense of (2.1), building an underdetermined interpolation quadratic model m $\in \mathcal{P}_n^2(\mathbb{R})$ can be viewed as finding $\alpha \in \mathbb{R}^{q+1}$ such that $\sum_{j=0}^{q} \alpha_j \phi_j(y^i) = f(y^i)$, for $i = 0, \dots, p$, or, equivalently,

$$\mathbf{M}(\phi, \mathcal{Y})\alpha = f(\mathcal{Y}), \tag{2.5}$$

where $\phi = \{\phi_0(x), \phi_1(x), \dots, \phi_q(x)\}$ is a basis for $\mathcal{P}_n^2(\mathbb{R})$,

$$\mathbf{M}(\phi, \mathcal{Y}) = \begin{bmatrix} \phi_0(y^0) & \phi_1(y^0) & \cdots & \phi_q(y^0) \\ \phi_0(y^1) & \phi_1(y^1) & \cdots & \phi_q(y^1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(y^p) & \phi_1(y^p) & \cdots & \phi_q(y^p) \end{bmatrix}, \text{ and } f(\mathcal{Y}) = \begin{bmatrix} f(y^0) \\ f(y^1) \\ \vdots \\ f(y^p) \end{bmatrix}. \tag{2.6}$$

Under the above conditions, the number of interpolation points is less than the dimension of the space of the polynomials $\mathcal{P}_n^2(\mathbb{R})$, that is, $|\mathcal{Y}| = p + 1 < q + 1 = dim(\mathcal{P}_n^2(\mathbb{R}))$. Therefore, the interpolation polynomials defined by (2.5) are no longer unique.

In this section, our goal is to build an underdetermined quadratic model, which approximately interpolates the function f over $\mathcal{Y} \subset \overline{B}(y^0, \delta)$, in the sense of Assumption 2.4. First, let us consider matrices

$$\mathbf{L}_s = \begin{bmatrix} (y^1 - y^0)^T \\ \vdots \\ (y^p - y^0)^T \end{bmatrix} = \begin{bmatrix} y_1^1 - y_1^0 & \dots & y_n^1 - y_n^0 \\ \vdots & \ddots & \vdots \\ y_1^p - y_1^0 & \dots & y_n^p - y_n^0 \end{bmatrix}, \text{ and } \hat{\mathbf{L}}_s = \frac{1}{\delta}\mathbf{L}_s. \tag{2.7}$$

In order to obtain the main error bounds for the underdetermined case, we need to assume some geometric conditions. In other words, we need matrix $\mathbf{L}_s$ to satisfy conditions similar to Assumption 2.3, and the Hessian of the model to be bounded. Such properties are given by Assumptions 2.8 and 2.9.

Assumption 2.8. *The matrix $\mathbf{L}_s \in \mathbb{R}^{p \times n}$ has full column rank, that is, $rank(\mathbf{L}_s) = n$, and there is a constant $\kappa_s > 0$ such that $\left\| \hat{\mathbf{L}}_s^\dagger \right\| \leq \kappa_s$, where $\hat{\mathbf{L}}_s^\dagger$ denotes the Moore-Penrose pseudo-inverse of the matrix $\hat{\mathbf{L}}_s$.*

Assumption 2.9. *The hessian of the model is bounded, that is, there is a constant $\kappa_H \geq 0$ such that $\|\mathbf{H}\| \leq \kappa_H$.*

In a similar way to what happens for determined models, Assumption 2.8 is satisfied if the sample set $\mathcal{Y}$ is Λ-poised in the linear regression sense [27, p. 69 and 83]. For the completeness of the text, we reproduce the definition of Λ-poisedness for regression models given by Conn et al. [27].

Definition 2.10. *[27, Definition 4.7] Let $\phi = \{\phi_0(x), \phi_1(x), \ldots, \phi_n(x)\}$, with $p > n$, be a basis in $\mathcal{P}_n^1$ and $\Lambda > 0$. A set $\mathcal{Y} = \{y^0, \ldots, y^p\} \subset \mathcal{X}$ is Λ-poised in $\overline{B}(y^0, \delta)$ (in the linear regression sense) if for each $x \in \overline{B}(y^0, \delta)$ there exists $\lambda(x) \in \mathbb{R}^{p+1}$, such that $\lambda(x)$ is the minimum l_2-norm solution of*

$$\sum_{i=0}^{p} \lambda_i(x)\phi_j(y^i) = \phi_j(x),$$

for all $j = 0, \ldots, n$, with $\|\lambda(x)\|_\infty \leq \Lambda$.

Since we are interested in quadratic models, we will not focus on linear regression models but on minimum Frobenius norm models, as we will see later. The following theorem establishes error bounds for the underdetermined model and its derivatives in $\mathcal{X} \cap \overline{B}(y^0, \delta)$.

Theorem 2.11. *Suppose that Assumptions 2.1, 2.2, 2.8 and 2.9 hold. Then, for all $x \in \mathcal{X} \cap \overline{B}(y^0, \delta)$,*

$$\|\nabla f(x) - \nabla \mathrm{m}(x)\| \leq 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\delta, \tag{2.8}$$

$$|f(x) - \mathrm{m}(x)| \leq \left(\frac{1}{2}(L + \kappa_H) + \kappa + 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\right)\delta^2. \tag{2.9}$$

Proof. The proof is strongly based in the ideas of [27, Theorem 5.4] and [72, Theorem 2]. For completeness, we present the proof as follows. In addition $\|\mathbf{H}\|$ and $\left\|\hat{\mathbf{L}}_s^\dagger\right\|$ were replaced by their respective bounds in equations (2.8) and (2.9).

Initially, note that for every sample point $y^j \in \mathcal{Y}$, and for all $x \in \mathcal{X} \cap \overline{B}(y^0, \delta)$,

$$\mathrm{m}(y^j) = \mathrm{m}(x) + \nabla \mathrm{m}(x)^T(y^j - x) + \frac{1}{2}(y^j - x)^T\mathbf{H}(y^j - x), \tag{2.10}$$

where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is a symmetric matrix. Given that the function f is continuously differentiable, by applying the Mean Value Theorem we have that for every $j = 0, 1, \ldots, q$, there exists $z^j \in [y^0, y^j] \subset \overline{B}(y^0, \delta)$ such that $f(y^j) = f(y^0) + \nabla f(z^j)^T(y^j - y^0)$. Therefore, by equation (2.10), we have that

$$\begin{aligned} \mathrm{m}(y^j) - f(y^j) = \mathrm{m}(x) + \nabla \mathrm{m}(x)^T(y^j - x) + \frac{1}{2}(y^j - x)^T \mathbf{H}(y^j - x) - f(y^0) \\ - \nabla f(z^j)^T(y^j - y^0), \text{ for } j = 0, 1, \ldots, q. \end{aligned} \tag{2.11}$$

By letting $j = 0$ in (2.11) and subtracting from (2.11), for $j = 1, \ldots, q$,

$$\begin{aligned} &\mathrm{m}(y^j) - f(y^j) - \big(\mathrm{m}(y^0) - f(y^0)\big) = \\ &\quad = \mathrm{m}(x) + \nabla \mathrm{m}(x)^T(y^j - x) + \frac{1}{2}(y^j - x)^T \mathbf{H}(y^j - x) - f(y^0) \\ &\quad\quad - \left(\mathrm{m}(x) + \nabla \mathrm{m}(x)^T(y^0 - x) + \frac{1}{2}(y^0 - x)^T \mathbf{H}(y^0 - x) - f(y^0)\right) \\ &\quad\quad - \nabla f(z^j)^T(y^j - y^0) \\ &\quad = \nabla \mathrm{m}(x)^T(y^j - y^0) + \frac{1}{2}(y^j - y^0)^T \mathbf{H}(y^j - y^0) - (y^j - y^0)^T \mathbf{H}(x - y^0) \\ &\quad\quad - \nabla f(z^j)^T(y^j - y^0), \end{aligned}$$

and, consequently,

$$\begin{aligned} \nabla \mathrm{m}(x)^T(y^j - y^0) = \nabla f(z^j)^T(y^j - y^0) + \left((x - y^0) - \frac{1}{2}(y^j - y^0)\right)^T \mathbf{H}(y^j - y^0) \\ + \mathrm{m}(y^j) - f(y^j) - \big(\mathrm{m}(y^0) - f(y^0)\big). \end{aligned}$$

Then, by subtracting $\nabla f(x)^T(y^j - y^0)$ from both sides of the equality, and by applying the Cauchy-Schwarz inequality, Assumption 2.1 and equation (2.2), we have that for all $x \in \mathcal{X} \cap \overline{B}(y^0, \delta)$, and for $j = 1, \ldots, q$,

$$\begin{aligned}
(\nabla \mathrm{m}(x) - \nabla f(x))^T (y^j - y^0) &= \mathrm{m}(y^j) - f(y^j) - \big(\mathrm{m}(y^0) - f(y^0)\big) \\
&\quad + \big(\nabla f(z^j) - \nabla f(x)\big)^T (y^j - y^0) \\
&\quad + \left((x - y^0) - \frac{1}{2}(y^j - y^0)\right)^T \mathbf{H}(y^j - y^0) \\
&\le \left|\mathrm{m}(y^j) - f(y^j)\right| + \left|\mathrm{m}(y^0) - f(y^0)\right| \\
&\quad + \left(\left\|x - y^0\right\| + \frac{1}{2}\left\|y^j - y^0\right\| \right) \|\mathbf{H}\| \left\|y^j - y^0\right\| \\
&\quad + \left\|\nabla f(z^j) - \nabla f(x)\right\| \left\|y^j - y^0\right\| \\
&\le 2\kappa\delta^2 + 2L\delta^2 + \left(\delta + \frac{\delta}{2}\right)\delta \|\mathbf{H}\| \\
&= 2\left(L + \kappa + \frac{3}{4}\|\mathbf{H}\|\right)\delta^2.
\end{aligned}$$

Therefore, it follows from Assumptions 2.8 and 2.9 that

$$\begin{aligned}
\|\nabla \mathrm{m}(x) - \nabla f(x)\| &= \left\|\hat{\mathbf{L}}_s^\dagger \hat{\mathbf{L}}_s \big(\nabla \mathrm{m}(x) - \nabla f(x)\big)\right\| \\
&\le \frac{1}{\delta}\left\|\hat{\mathbf{L}}_s^\dagger\right\| \left\|\mathbf{L}_s\big(\nabla \mathrm{m}(x) - \nabla f(x)\big)\right\| \\
&\le \frac{\sqrt{p}}{\delta}\left\|\hat{\mathbf{L}}_s^\dagger\right\| \left\|\mathbf{L}_s\big(\nabla \mathrm{m}(x) - \nabla f(x)\big)\right\|_\infty \\
&\le 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\delta.
\end{aligned} \tag{2.12}$$

Lastly, given that for all $x \in \mathcal{X} \cap \overline{B}(y^0, \delta)$,

$$\mathrm{m}(x) = \mathrm{m}(y^0) + \nabla \mathrm{m}(y^0)^T (x - y^0) + \frac{1}{2}(x - y^0)^T \mathbf{H}(x - y^0),$$

by applying the Cauchy-Schwarz inequality, Assumptions 2.8 and 2.9, equations (2.2) and (2.12), we have that

$$\begin{aligned}
|f(x) - \mathrm{m}(x)| &= \left| f(x) - \mathrm{m}(y^0) + \nabla \mathrm{m}(y^0)^T (x - y^0) + \frac{1}{2}(x - y^0)^T \mathbf{H}(x - y^0) \right| \\
&\le \left|f(x) - f(y^0) - \nabla f(y^0)^T (x - y^0)\right| + \frac{1}{2}\|\mathbf{H}\| \left\|x - y^0\right\|^2 \\
&\quad + \left|f(y^0) - \mathrm{m}(y^0)\right| + \left\|\nabla f(y^0) - \nabla \mathrm{m}(y^0)\right\| \left\|x - y^0\right\| \\
&\le \frac{1}{2}L\delta^2 + \frac{1}{2}\kappa_H\delta^2 + \kappa\delta^2 + 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\delta^2 \\
&= \left(\frac{1}{2}(L + \kappa_H) + \kappa + 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\right)\delta^2,
\end{aligned}$$

which concludes the proof. □

2.3.1 Minimum Frobenius norm models

To establish a bound for the norm of the Hessian of the quadratic model, we must specify its construction. In this sense, we will reduce the degree of freedom in choosing the model, assuming that it is the minimum Frobenius norm model. We will also need to define what is a relaxed minimum Frobenius norm model.

Let $\overline{\phi}$ be the natural basis of $\mathcal{P}_n^2(\mathbb{R})$ and consider its split into the sets $\overline{\phi}_L = \{1, x_1, \ldots, x_n\}$ and $\overline{\phi}_Q = \left\{\frac{1}{2}x_1^2, x_1x_2, \ldots, \frac{1}{2}x_n^2\right\}$. Then we can write the underdetermined interpolation model as $\mathrm{m}(x) = \alpha_L^T\overline{\phi}_L(x) + \alpha_Q^T\overline{\phi}_Q(x)$, where α_L and α_Q are appropriate parts of the vector of coefficients α. As in [27, p. 80-81], we can define the minimum Frobenius norm solution α^{mfn} as the solution of the optimization problem,

$$\begin{array}{ll} \min & \frac{1}{2}\left\|\alpha_Q\right\|^2 \\ \text{s.t.} & \mathbf{M}(\overline{\phi}_L, \mathcal{Y})\alpha_L + \mathbf{M}(\overline{\phi}_Q, \mathcal{Y})\alpha_Q = f(\mathcal{Y}), \end{array} \tag{2.13}$$

in the variables α_L and α_Q, where $\mathcal{Y}$ is the set of interpolation points and the matrices $\mathbf{M}(\overline{\phi}_L, \mathcal{Y})$ and $\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})$ are submatrices of $\mathbf{M}(\overline{\phi}, \mathcal{Y})$, given in (2.6), whose columns correspond to the elements of $\overline{\phi}_L$ and $\overline{\phi}_Q$, respectively.

It is important to point out that the condition of existence and uniqueness of the minimum Frobenius norm model is that the matrix defined by

$$\mathbf{F}(\overline{\phi}, \mathcal{Y}) = \begin{bmatrix} \mathbf{M}(\overline{\phi}_Q, \mathcal{Y})\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})^T & \mathbf{M}(\overline{\phi}_L, \mathcal{Y}) \\ \mathbf{M}(\overline{\phi}_L, \mathcal{Y})^T & 0 \end{bmatrix}$$

is nonsingular. Note that $\mathbf{F}(\overline{\phi}, \mathcal{Y})$ is nonsingular if and only if $\mathbf{M}(\overline{\phi}_L, \mathcal{Y})$ has a full column rank and $\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})^T$ is positive definite on the nullspace of $\mathbf{M}(\overline{\phi}_L, \mathcal{Y})^T$, this last condition being guaranteed if $\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})$ has full row rank [27, p. 81].

Definition 2.12. *[27, p. 81] Let $\overline{\phi}$ be the natural basis of monomials of $\mathcal{P}_n^2(\mathbb{R})$. A sample set $\mathcal{Y} = \{y^0, y^1, \ldots, y^p\}$ is said to be poised in the minimum Frobenius norm sense if the matrix $\mathbf{F}(\overline{\phi}, \mathcal{Y})$ is nonsingular.*

Note that poisedness in the minimum Frobenius norm sense implies poisedness in the linear interpolation or regression senses and, as a result, poisedness for underdetermined quadratic interpolation in the minimum norm sense [27, p. 81].

Remark 2.13. *As we saw earlier, if $\mathcal{Y}$ is poised in the minimum Frobenius norm sense, then $\mathbf{F}(\overline{\phi}, \mathcal{Y})$ is nonsingular and hence $\mathbf{M}(\overline{\phi}_L, \mathcal{Y})$ has a full column rank. So, as*

$$\mathbf{M}(\overline{\phi}_L, \mathcal{Y}) = \begin{bmatrix} 1 & 0 \\ e & \mathbf{I}_p \end{bmatrix} \begin{bmatrix} 1 & {y^0}^T \\ 0 & \mathbf{L}_s \end{bmatrix} = \mathbf{E}^{-1} \begin{bmatrix} 1 & {y^0}^T \\ 0 & \mathbf{L}_s \end{bmatrix}$$

where $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^p$, and $\mathbf{I}_p \in \mathbb{R}^{p \times p}$ is the identity matrix. Since $\mathbf{E}$ is an elementary nonsingular matrix and $\mathbf{M}(\overline{\phi}_L, \mathcal{Y})$ has full column rank, it follows that $\mathbf{L}_s$ has full column rank, which partially satisfies Assumption 2.8.

Therefore, we will assume that the following hypothesis holds.

Assumption 2.14. *The set of sample points $\mathcal{Y} = \{y^0, y^1, \ldots, y^p\}$ is poised in the minimum Frobenius norm sense on $\overline{B}(y^0, \delta)$.*

Consider the following auxiliary lemma.

Lemma 2.15. *There is a number $\sigma_\infty > 0$ such that for any choice of v satisfying $\|v\|_\infty = 1$, there is $z \in \overline{B}(0,1)$ such that $\left|v^T \phi(z)\right| \geq \sigma_\infty$. If, in addition, $v^T \overline{\phi}(x)$ is a quadratic polynomial and $\overline{\phi}$ is the natural basis in $\mathcal{P}_n^2(\mathbb{R})$, then*

$$\max_{x \in \overline{B}(0,1)} \left|v^T \overline{\phi}(x)\right| \geq \frac{1}{4}.$$

If $\overline{\phi}$ is the natural basis of $\mathcal{P}_n^1(\mathbb{R})$, then,

$$\max_{x \in \overline{B}(0,1)} \left|v^T \overline{\phi}(x)\right| \geq 1.$$

Proof. See [27, Lemma 3.10] and [27, Lemma 6.7]. □

Remark 2.16. *Let $\overline{\phi}$ be the natural basis of $\mathcal{P}_n^1(\mathbb{R})$. Given $\overline{v} \in \mathbb{R}^{n+1}$ with $\|\overline{v}\| = 1$, there are $\beta \in (0, \sqrt{n+1})$ and $v \in \mathbb{R}^{n+1}$ such that $\|v\|_\infty = 1$ and $v = \beta \overline{v}$. Thus, by Lemma 2.15, we have that*

$$\max_{x \in \overline{B}(0,1)} \left|\overline{v}^T \overline{\phi}(x)\right| = \max_{x \in \overline{B}(0,1)} \frac{1}{\beta} \left|v^T \overline{\phi}(x)\right| \geq \frac{1}{\sqrt{n+1}} \max_{x \in \overline{B}(0,1)} \left|v^T \overline{\phi}(x)\right| \geq \frac{1}{\sqrt{n+1}}.$$

Equation (2.13) is strongly related with the exact interpolation condition (2.1). In order to relax such condition we will first define minimum Frobenius norm Lagrange polynomials.

Definition 2.17. *[27, Definition 5.5] Given a set $\mathcal{Y}$ of $p+1$ interpolation points, the set of $p+1$ polynomials $\ell_j(x) = \alpha_j^T \overline{\phi}(x)$, $j = 0, \ldots, p$, is called set of minimum Frobenius norm Lagrange polynomials for $\overline{\phi}$, if $\alpha_j = ((\alpha_j)_L, (\alpha_j)_Q)$ is the solution of*

$$\begin{array}{ll} \min & \frac{1}{2}\left\|\alpha_Q\right\|^2 \\ s.\ t. & \mathbf{M}(\overline{\phi}_L, \mathcal{Y})\alpha_L + \mathbf{M}(\overline{\phi}_Q, \mathcal{Y})\alpha_Q = e_{j+1}, \end{array}$$

where e_j is the j-th canonical vector in $\mathbb{R}^{p+1}$.

By comparing the KKT conditions of Definition 2.17 and equation (2.13) it is possible to show that the minimum Frobenius norm polynomial that interpolates f in $\mathcal{Y}$ can be defined as $\mathrm{m}(x) = \sum_{j=1}^{p+1} \ell_j(x) f(y^j)$. So, in order to relax (2.13), we will say that our relaxed minimum Frobenius norm polynomials are given by

$$\mathrm{m}(x) = \sum_{j=1}^{p+1} \ell_j(x)\gamma_j \tag{2.14}$$

where γ_j, $j = 1, \ldots, p+1$, are chosen such that Assumption 2.4 holds. One can think that m was obtained by solving (2.13) with $f(y^j)$ replaced by γ_j for all $j = 0, \ldots, p$.

The theorems that follow are the main contribution of this chapter. Their establish bounds for $\|\nabla^2 m(x)\|$ and $\left\|\hat{\mathbf{L}}_s^\dagger\right\|$ under the hypothesis of well poised interpolation sets and, therefore, Assumptions 2.8 and 2.9 can be removed from Theorem 2.11. Using a partial definition from [27], we first define what we mean as well poised interpolation sets for minimum Frobenius norm quadratic models.

Definition 2.18. *[27, Definition 5.6] Let $\Lambda > 0$ and $B \subseteq \mathbb{R}^n$ be given. Let $\overline{\phi}$ be the natural basis of monomials of $\mathcal{P}_n^2(\mathbb{R})$. A poised set $\mathcal{Y}$ is said to be Λ-poised in B (in the minimum Frobenius norm sense) if and only if for any $x \in B$, the solution $\lambda(x) \in \mathbb{R}^{p+1}$ of*

$$\begin{array}{ll} \min & \frac{1}{2}\left\|\mathbf{M}(\overline{\phi}_Q, \mathcal{Y})^T \lambda(x) - \overline{\phi}_Q(x)\right\|^2 \\ s.t. & \mathbf{M}(\overline{\phi}_L, \mathcal{Y})^T \lambda(x) = \overline{\phi}_L(x) \end{array}$$

is such that $\|\lambda(x)\|_\infty \leq \Lambda$.

Note that Definition 2.18 does not need to be relaxed. We are now ready to provide the necessary bounds in terms of the quality of the interpolation set, the Lipschitz constant and the dimension of the problem.

Theorem 2.19. *Suppose that Assumptions 2.1 and 2.14 hold. Assume that the set* $\mathcal{Y} = \{y^0, y^1, \ldots, y^p\}$ *is* Λ*-poised in* $\overline{B}(y^0,\delta)$ *in the minimum Frobenius norm sense and* $\delta_{\max} > 0$ *is an upper bound for* δ*. Then,*

$$\left\|\nabla^2 \mathrm{m}(x)\right\| \leq \left(\kappa + \frac{L}{2}\right) \frac{4\Lambda(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2},$$

where $c(\delta_{\max}) = \min\{1, 1/\delta_{\max}, 1/\delta_{\max}^2\}$.

Proof. This proof follows the proof of [27, Theorem 5.7]. Assume, without loss of generality, that $y^0 = 0$. By Lemma 2.15, the definition of Λ-poisedness and arguments very similar to [27, Theorem 5.7] we have that

$$\left\|\nabla^2 \ell_j(x)\right\| \leq \sqrt{2(q+1)} \frac{4\Lambda}{\delta^2 c(\delta_{\max})^2}, \quad j = 0, \ldots, p, \tag{2.15}$$

where $\delta_{\max}$ and $c(\delta_{\max})$ were defined by the theorem.

Now, let us consider the function $\hat{f}(x) = f(x) - f(y^0) - \nabla f(y^0)^T(x - y^0)$. Note that $\hat{f}(y^0) = 0$, $\nabla \hat{f}(y^0) = 0$ and the Hessian remains unchanged. In addition, if $\mathrm{m}(x)$ is a relaxed minimum Frobenius norm model for f over the set $\mathcal{Y}$, then it is easy to see that $\hat{\mathrm{m}}(x) = \mathrm{m}(x) - f(y^0) - \nabla f(y^0)^T(x - y^0)$ also satisfies Assumption 2.4 for $\hat{f}$ and the points in $\mathcal{Y}$. Therefore, we can assume without loss of generality that $f(y^0) = 0$ and $\nabla f(y^0) = 0$. Thus, we have that $|f(x)| = \left|f(x) - f(y^0) - \nabla f(y^0)^T(x - y^0)\right| \leq (L/2)\delta^2$, for all $x \in B(y^0,\delta)$ and, by (2.14), we can prove that $|\gamma_j| \leq (\kappa + L/2)\delta^2, \quad j = 0, \ldots, p$. By using the definition of m in (2.14) and (2.15), we finally get that

$$\left\|\nabla^2 \mathrm{m}(x)\right\| \leq \sum_{j=0}^{p} |\gamma_j| \left\|\nabla^2 \ell_j(x)\right\| \leq \left(\kappa + \frac{L}{2}\right) \frac{4\Lambda(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2},$$

and we conclude the proof. □

In order to help us prove the following theorem, consider the auxiliary lemmas.

Lemma 2.20. *Let* $\mathbf{M} \in \mathbb{R}^{m \times n}$ *be a matrix partitioned as* $\mathbf{M} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{B} & \mathbf{C} \end{bmatrix}$. *Then the Moore-Penrose pseudoinverse of* $\mathbf{M}$ *is given by*

$$\mathbf{M}^{\dagger} = \begin{bmatrix} \mathbf{K}^{\dagger}\mathbf{A}^T & \mathbf{K}^{\dagger}\mathbf{B}^T \\ 0 & \mathbf{C}^{\dagger} \end{bmatrix},$$

where $\mathbf{K} = \mathbf{A}^T\mathbf{A} + \mathbf{B}^T\mathbf{B}$*, if and only if* $\mathbf{B}^T\mathbf{C} = 0$.

Proof. See [43, Lemma 2]. □

Lemma 2.21. *Let w be a right-singular vector of matrix $\mathbf{A}$ corresponding to its largest singular value. Then, for any vector r of the appropriate size, $\|\mathbf{A}r\| \geq |w^T r| \, \|\mathbf{A}\|$.*

Proof. See [27, Lemma 3.13]. □

Theorem 2.22. *If $\mathcal{Y} = \{y^0, y^1, \ldots, y^p\}$ is a Λ-poised set in $\overline{B}(y^0, \delta)$ in the sense of the minimum Frobenius norm, then $\left\|\hat{\mathbf{L}}_s^\dagger\right\| \leq \Lambda\sqrt{2(n+1)}(p+1)$.*

Proof. It is known that Λ-poisedness does not depend on the scale of the sample set and it is invariant with respect to shifts [27]. Therefore, let us consider the set $\hat{\mathcal{Y}} = \{0, (y^1 - y^0)/\delta, \ldots, (y^p - y^0)/\delta\}$ which is Λ-poised in $\overline{B}(0,1)$, and the matrices given by $\hat{\mathbf{M}}_L = \mathbf{M}(\overline{\phi}_L, \hat{\mathcal{Y}})$,

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ -e & \mathbf{I}_p \end{bmatrix}, \mathbf{E}^{-1} = \begin{bmatrix} 1 & 0 \\ e & \mathbf{I}_p \end{bmatrix}, \text{ and } \mathbf{Q} = \mathbf{E}\hat{\mathbf{M}}_L = \begin{bmatrix} 1 & 0 \\ 0 & \hat{\mathbf{L}}_s \end{bmatrix},$$

where $\hat{\mathbf{M}}_L \in \mathbb{R}^{(p+1)\times(n+1)}$, $\mathbf{E} \in \mathbb{R}^{(p+1)\times(p+1)}$, $\mathbf{E}^{-1} \in \mathbb{R}^{(p+1)\times(p+1)}$, $\mathbf{Q} \in \mathbb{R}^{(p+1)\times(n+1)}$, $e = [1,1,\ldots,1]^T \in \mathbb{R}^p$, $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ is the identity matrix and $\hat{\mathbf{L}}_s$ was defined in (2.7). Note that the Moore-Penrose pseudoinverse of $\mathbf{Q}$ is given by $\mathbf{Q}^\dagger = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T \in \mathbb{R}^{(n+1)\times(p+1)}$. Thus, by Lemma 2.20,

$$\|\mathbf{Q}^\dagger\| = \left\| \begin{bmatrix} 1 & 0 \\ 0 & \hat{\mathbf{L}}_s^\dagger \end{bmatrix} \right\| = \max\left\{1, \left\|\hat{\mathbf{L}}_s^\dagger\right\|\right\} \geq \left\|\hat{\mathbf{L}}_s^\dagger\right\|. \tag{2.16}$$

By Definition 2.18, for every $x \in \overline{B}(0,1) \subset \mathbb{R}^n$ there exists $\lambda(x) \in \mathbb{R}^{p+1}$, with $\|\lambda(x)\|_\infty \leq \Lambda$, such that $\hat{\mathbf{M}}_L^T\lambda(x) = \overline{\phi}_L(x)$. Since $\mathbf{E}$ is a non-singular matrix, we have that for each $x \in \overline{B}(0,1)$,

$$\begin{aligned} \hat{\mathbf{M}}_L^T\lambda(x) = \overline{\phi}_L(x) &\iff \hat{\mathbf{M}}_L^T\left(\mathbf{E}^T\mathbf{E}^{-T}\right)\lambda(x) = \overline{\phi}_L(x) \\ &\iff \left(\hat{\mathbf{M}}_L^T\mathbf{E}^T\right)\left(\mathbf{E}^{-T}\lambda(x)\right) = \overline{\phi}_L(x) \\ &\iff \left(\mathbf{E}\hat{\mathbf{M}}_L\right)^T\left(\mathbf{E}^{-T}\lambda(x)\right) = \overline{\phi}_L(x) \\ &\iff \mathbf{Q}^T\left(\mathbf{E}^{-T}\lambda(x)\right) = \overline{\phi}_L(x) \\ &\iff \mathbf{E}^{-T}\lambda(x) = \mathbf{Q}^{\dagger^T}\overline{\phi}_L(x). \end{aligned} \tag{2.17}$$

Note that $\|\lambda(x)\|_\infty \leq \Lambda$ and $\mathbf{E}^{-T}\lambda(x) = \left[\sum_{j=0}^p [\lambda(x)]_j, \lambda(x)^T\right]^T$. So, by (2.17),

$$\begin{aligned}
\left\|\mathbf{Q}^{\dagger^T}\overline{\phi}_L(x)\right\| &= \left\|\mathbf{E}^{-T}\lambda(x)\right\| \\
&= \sqrt{\left(\sum_{j=0}^p [\lambda(x)]_j\right)^2 + \sum_{j=1}^p [\lambda(x)]_j^2} \\
&\leq \sqrt{\left(\sum_{j=0}^p |[\lambda(x)]_j|\right)^2 + \sum_{j=1}^p |[\lambda(x)]_j|^2} \\
&\leq \sqrt{\big((p+1)\Lambda\big)^2 + p\Lambda^2} < \sqrt{2(p+1)^2\Lambda^2} \\
&= \Lambda\sqrt{2}(p+1).
\end{aligned} \tag{2.18}$$

Let $\overline{v} \in \mathbb{R}^{n+1}$ be a singular right unit vector of $\mathbf{Q}^{\dagger^T}$ associated with the largest singular value σ_1, and $\overline{x} \in \mathbb{R}^n$ the maximizer of $\left|\overline{v}^T\overline{\phi}_L(x)\right|$ in $\overline{B}(0,1)$. Using Lemma 2.21 and the fact that $\left\|\mathbf{Q}^\dagger\right\| = \sigma_1$, we have that $\left\|\mathbf{Q}^\dagger\overline{\phi}_L(\overline{x})\right\| \geq |\overline{v}^T\overline{\phi}_L(\overline{x})| \left\|\mathbf{Q}^{\dagger^T}\right\|$. Then, from Lemma 2.15 and Remark 2.16,

$$\left\|\mathbf{Q}^{\dagger^T}\overline{\phi}_L(\overline{x})\right\| \geq \left|\overline{v}^T\overline{\phi}_L(\overline{x})\right| \left\|\mathbf{Q}^\dagger\right\| = \max_{x\in\overline{B}(0,1)} \left|\overline{v}^T\overline{\phi}_L(x)\right| \left\|\mathbf{Q}^\dagger\right\| \geq \frac{1}{\sqrt{n+1}} \left\|\mathbf{Q}^\dagger\right\|.$$

Thus, from the previous inequality, (2.16) and (2.18), it follows that $\left\|\hat{\mathbf{L}}_s^\dagger\right\| \leq \left\|\mathbf{Q}^\dagger\right\| \leq \Lambda\sqrt{2(n+1)}(p+1)$, and we complete the proof. □

2.4 Summary of complexity constants

This section is dedicated to collecting the error bounds described in Sections 2.2 and 2.3, for determined and underdetermined "inexact" interpolation models, in the sense of Assumption 2.4. Table 2.1 organizes and summarizes these constants to make their use practical.

Model Type	**Error**	**Error bound**	Ref.
Linear determined	$\lvert \mathrm{m}(x) - f(x) \rvert$	$\left(\frac{1}{2}L + \kappa + \left(\frac{1}{2}L + 2\kappa\right)\Lambda n\right)\delta^2$	[72]
	$\lVert \nabla \mathrm{m}(x) - \nabla f(x) \rVert$	$\left(L + \left(\frac{1}{2}L + 2\kappa\right)\Lambda n\right)\delta$	[72]
	$\left\lVert \nabla^2 \mathrm{m}(x) \right\rVert$	0	[72]
Quad. determined	$\lvert \mathrm{m}(x) - f(x) \rvert$	$\left(\frac{1}{2}L + \kappa + 4\Lambda\sqrt{q(q+1)^3}\left(2 + 3\sqrt{2}\right)(\kappa + L)\right)\delta^2$	[72]
	$\lVert \nabla \mathrm{m}(x) - \nabla f(x) \rVert$	$\left(8\Lambda\sqrt{q(q+1)^3}\left(1 + \sqrt{2}\right)(\kappa + L)\right)\delta$	[72]
	$\left\lVert \nabla^2 \mathrm{m}(x) \right\rVert$	$8\Lambda\sqrt{2q(q+1)^3}(\kappa + L)$	[72]
Quad. underdetermined	$\lvert \mathrm{m}(x) - f(x) \rvert$	$\left(\frac{1}{2}\left(L + \kappa_H\right) + \kappa + 2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\right)\delta^2$	Thm. 2.11
	$\lVert \nabla \mathrm{m}(x) - \nabla f(x) \rVert$	$2\kappa_s\sqrt{p}\left(L + \kappa + \frac{3}{4}\kappa_H\right)\delta$	Thm. 2.11
Min. Frobenius norm	$\lvert \mathrm{m}(x) - f(x) \rvert$	$\left(\frac{1}{2}\left(L + \left(\kappa + \frac{L}{2}\right)\Lambda\frac{4(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2}\right) + \kappa + \right.$ $\left. + 2\Lambda\sqrt{2p(n+1)}(p+1)\left(L + \kappa + \left(\kappa + \frac{L}{2}\right)\Lambda\frac{3(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2}\right)\right)\delta^2$	Thm. 2.11, 2.19, 2.22
	$\lVert \nabla \mathrm{m}(x) - \nabla f(x) \rVert$	$2\Lambda\sqrt{2p(n+1)}(p+1)\left(L + \kappa + \left(\kappa + \frac{L}{2}\right)\Lambda\frac{3(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2}\right)\delta$	Thm. 2.11, 2.19, 2.22
	$\left\lVert \nabla^2 \mathrm{m}(x) \right\rVert$	$\left(\kappa + \frac{L}{2}\right)\Lambda\frac{4(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2}$	Thm. 2.19

Table 2.1: Error bounds for linear and quadratic interpolation models under Assumption 2.4.

Chapter 3

A derivative-free trust-region LOVO algorithm

This chapter is organized as follows. Section 3.1 provides a short overview about low order-value optimization problems and derivative-free optimization methods. We dedicate part of the section to reviewing the central ideas of some recent works in the area, in particular, on methods capable of solving problems with characteristics similar to LOVO problems. In Section 3.2, we introduce the derivative-free trust-region framework and our algorithm for LOVO problems. Global convergence analysis results are presented and discussed in Section 3.3, as well as an interpretation from the perspective of the classical theory of LOVO problems. Finally, in Section 3.4, we study the worst-case complexity of our algorithm.

3.1 Short overview

Let us consider the following constrained nonlinear optimization problem

$$\min_{x\in\Omega} f_{min}(x) = \min_{x\in\Omega} \min\{f_1(x), \dots, f_r(x)\}, \tag{3.1}$$

where $\Omega \subset \mathbb{R}^n$ is a nonempty closed convex set and $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \dots, r$, are continuously differentiable black-box functions. Although we consider that each function that composes the objective function is smooth, we assume that there is no information available about its first and second-order derivatives. As we saw earlier, the problem (3.1) is known as the Low Order-Value Optimization problem (LOVO), for which extensive theory and applications exist [2, 47].

The main contribution of this chapter is the development and analysis of the first derivative-free optimization algorithm designed for LOVO problems. Note that the LOVO problem can be understood as a particular case of nonsmooth composite optimization. In this sense, Garmanjani et al. [35] and Grapiglia et al. [37], propose model-based derivative-free algorithms for objective functions of the form $f + h \circ g$, where f and g are smooth, and h is convex. However, such methods are not compatible with the specificities of the LOVO problem since the minimum function is not convex. Furthermore, the construction of the models involves the evaluation of $h \circ g$ at each one of the points in the sample set, which can impair the performance of the method, especially in cases where evaluations of the functions g or h have a high computational cost. Recently, Larson et al. [46] propose a manifold sampling algorithm for minimizing nonsmooth compositions $g \circ f$, where g is nonsmooth, f is smooth with no information about its derivatives, and $g \circ f$ is a continuous selection [46, Definition 1.1]. From a theoretical point of view, the algorithm generates a sequence of iterates that converge to a Clarke stationary point (see [16]). However, [48] shows that the concept of Clarke stationarity is not sufficiently satisfactory for LOVO problems, which makes it necessary to adopt methods that are provably to converge to points that satisfy stronger optimality conditions, such as [48, Theorem 6.1].

It is worth mentioning that [1] presents a trust-region LOVO algorithm with local and global convergence. In this sense, the proposed method is an improvement as we allow general quadratic models and closed convex constraints. Finally, Hough and Roberts [42] propose a model-based derivative-free method for black-box optimization problems subject to convex constraints, in an approach similar to the trust-region structure presented in [17] but with another criticality measure. However, they assume that the objective function is smooth, which may not be valid for LOVO problems. In this work, the authors also generalize the notions of fully linear models and Λ-poised interpolation sets discussed in Chapter 2 (see [27]) to arbitrary convex sets. This new approach allows us to solve a well-known problem in derivative-free optimization, namely in cases where it might be impossible to build fully linear models using only feasible points. However, the authors do not propose extensions to other forms of poisedness, such as those allowed in our method.

3.2 Framework and algorithm

In this section, we present our algorithm for solving the LOVO problem (3.1). In order to help us with the theoretical results exposed in this text, consider the following definition.

Definition 3.1. [15, Definition 1] Given a feasible point $x \in \Omega$ we define

$$\mathcal{I}_{min}(x) = \{i \in \{1, \dots, q\} \mid f_i(x) = f_{min}(x)\}.$$

For convenience of notation, we will denote the set of component function indices by $\mathcal{I} := \{1, 2, \dots, q\}$. We consider the following assumptions on the objective function.

Assumption 3.2. *The function $f_i : \mathbb{R}^n \to \mathbb{R}$, for $i \in \mathcal{I}$, is continuously differentiable and its gradient $\nabla f_i : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous, with constant $L_i > 0$, in a sufficiently large open bounded domain $\mathcal{X} \subset \mathbb{R}^n$.*

Assumption 3.3. *The function $f_i : \mathbb{R}^n \to \mathbb{R}$, for $i \in \mathcal{I}$, is bounded below in $\Omega \subset \mathbb{R}^n$, i.e., there is a constant $M_i \in \mathbb{R}$ such that $f_i(x) \geq M_i$, for all $x \in \Omega$.*

Remark 3.4. *If Assumption 3.3 holds and $M := \min_{i \in \mathcal{I}}\{M_i\}$, then the function f_{min} is bounded below in Ω by the constant M.*

Our algorithm is based on the trust-region frameworks of [1, 72]. At each iteration $k \in \mathbb{N}$, we consider the current iterate $x_k \in \Omega$, the associated index $i_k \in \mathcal{I}_{min}(x_k)$ and the quadratic model for f_{i_k}

$$\mathrm{m}_k(x) = \overline{\mathrm{m}}_k(x_k + d) = b_k + \mathbf{g}_k^T d + \frac{1}{2} d^T \mathbf{H}_k d, \tag{3.2}$$

where $d = x - x_k \in \mathbb{R}^n$, $b_k \in \mathbb{R}$, $\mathbf{g}_k \in \mathbb{R}^n$ and $\mathbf{H}_k \in \mathbb{R}^{n \times n}$ is a symmetric matrix. In [1], a similar approach is taken with $\mathbf{g}_k = \nabla f_{i_k}(x_k)$ and $\mathbf{H}_k = \nabla^2 f_{i_k}(x_k)$. Note that only one index $i_k \in \mathcal{I}_{min}(x_k)$ is chosen at iteration $k \in \mathbb{N}$. Thus, we will omit the indication of such index in the model definition and other expressions that depend on the choice of index.

We also follow the practical ideas of Powell [58, 59], further developed in [72], of using two radii: δ_k is related to the quality of the model, and Δ_k is associated with the trust-region. Assumption 3.5 below defines what "quality of the model" means and is a weaker assumption than “fully linear models” of [27], since it does not especify how to build such models.

Assumption 3.5. *For all $k \in \mathbb{N}$, $\|\nabla f_{i_k}(x) - \nabla \mathrm{m}_k(x)\| \le \kappa_g \delta_k$, for all $x \in \mathcal{X} \cap \overline{B}(x_k, \delta_k)$ and some constant $\kappa_g > 0$ independent of x and k.*

We define the stationarity measure at x_k for the problem of minimizing m_k over the set Ω by

$$\pi_k = \|P_\Omega(x_k - \mathbf{g}_k) - x_k\|, \tag{3.3}$$

where P_Ω denotes the orthogonal projection onto Ω, which exists because it is a closed convex set [25, 62]. In our context, given $i \in \mathcal{I}$, we say that a point $x_* \in \Omega$ is stationary for the problem $\min_{x \in \Omega} f_i(x)$ when $\|P_\Omega(x_* - \nabla f_i(x_*)) - x_*\| = 0$ [25, Section 12.1.4].

At iteration $k \in \mathbb{N}$, the candidate point $x_k + d_k \in \Omega$ is computed, where d_k is the approximate solution of the following trust-region subproblem

$$\begin{array}{rl} \text{minimize} & \mathrm{m}_k(x_k + d) \\ \text{subject to} & x_k + d \in \Omega \\ & \|d\| \le \Delta_k. \end{array} \tag{3.4}$$

If the model is accurate for f_{i_k}, we expect that $x_k + d_k$ may also decrease f_{min}, since $x_k + d_k \in \Omega$ and $f_{min}(x_k + d_k) \le f_{i_k}(x_k + d_k) \le f_{i_k}(x_k)$. By “approximate solution” we mean that $x_k + d_k$ must satisfy a sufficient decrease condition of m_k, given by Assumption 3.6.

Assumption 3.6. *The approximate solution of* (3.4) *satisfies the sufficient decrease condition*

$$\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k) \ge \theta \pi_k \min\left\{ \frac{\pi_k}{1 + \|\mathbf{H}_k\|}, \Delta_k, 1 \right\},$$

for some constant $\theta > 0$ independent of k.

Conditions of this type are well-known in trust-region strategies, and were used in different contexts by several authors, as we can see in [17, 72], [5, Lemma 11.3], [25, Assumption AA.1], [27, Theorem 10.1], [42, Assumption 3.3], [55, Lemma 4.5], and [70, Section 3.1.4].

We accept step $x_k + d_k$ when the ratio between actual and predicted reductions

$$\rho_k = \frac{f_{min}(x_k) - f_{min}(x_k + d_k)}{\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)} \tag{3.5}$$

is greater than or equal to a fixed constant $\eta > 0$. In this case, the new iterate becomes $x_{k+1} = x_k + d_k$, the model is updated and the trust-region radius Δ_k is possibly increased. Otherwise, we reject the step and Δ_k is decreased.

We present now our derivative-free trust-region algorithm for LOVO problems, without specification about the model update or the solution of subproblem (3.4). Our algorithm is based on the ideas discussed by Andreani et al. [1] and on the algorithms proposed by Conejo et al. [17] and Verdério et al. [72].

Algorithm 1: DERIVATIVE-FREE TRUST-REGION LOVO ALGORITHM

```
Input: $x_0 \in \Omega$, $\beta > 0$, $0 < \delta_0 \leq \Delta_0$, $0 < \tau_1 \leq \tau_2 < 1 \leq \tau_3 \leq \tau_4$, $\eta_1 \in (0,1)$, $0 \leq \eta < \eta_1 \leq \eta_2$, $\Gamma = 0$,
       $1 \leq \Gamma_{\max} \in \mathbb{N}$, $i_0 \in \mathcal{I}_{min}(x_0)$.
1  for $k = 0, 1, \ldots$ do
2      Construct the model $\mathrm{m}_k$.
3      if $\delta_k > \beta\pi_k$ then                                    /* Criticality Phase */
4          Let $x_{k+1} = x_k$, $i_{k+1} = i_k$, $\rho_k = 0$, $d_k = 0$, $\delta_{k+1} = \tau_1\delta_k$, and choose $\Delta_{k+1} \in [\tau_1\Delta_k, \tau_2\Delta_k]$.
5      else
6          Find an approximate solution $d_k$ for (3.4) that satisfies Assumption 3.6.
7          Compute $\rho_k$ by (3.5).
8          if $\rho_k \geq \eta$ then                                  /* Step Acceptance Phase */
9              Let $x_{k+1} = x_k + d_k$, and choose $i_{k+1} \in \mathcal{I}_{min}(x_{k+1})$.
10         else
11             Let $x_{k+1} = x_k$, and $i_{k+1} = i_k$.
12         end
13         if $\rho_k \geq \eta_1$ then $\Gamma = 0$ end
14         if $\rho_k \geq \eta$ and $i_{k+1} \neq i_k$ and $\Gamma \leq \Gamma_{\max}$ then     /* Radii Adjustment Phase */
15             Let $\delta_{k+1} = \tau_4\delta_k$, $\Delta_{k+1} = \tau_4\Delta_k$, and $\Gamma = \Gamma + 1$.
16         else                                                         /* Radii Update Phase */
17             Let
                   $$\delta_{k+1} = \begin{cases} \tau_1\delta_k, \text{ if } \rho_k < \eta_1; \\ \tau_3\delta_k, \text{ if } \rho_k > \eta_2 \text{ and } \|d_k\| = \Delta_k; \\ \delta_k, \text{ otherwise;} \end{cases}$$
               and
                   $$\Delta_{k+1} = \begin{cases} \tau_1\Delta_k, \text{ if } \rho_k < \eta_1; \\ \tau_3\Delta_k, \text{ if } \rho_k > \eta_2 \text{ and } \|d_k\| = \Delta_k; \\ \Delta_k, \text{ otherwise.} \end{cases}$$
18         end
19     end
20 end
```

The general idea of the algorithm is given below:

- The algorithm allows freedom in the choice of m_k as long as it satisfies Assumption 3.5. However, its practical efficiency greatly relies in the way it is implemented, as discussed in Section 4.2.

- When δ_k is large with respect to π_k, we cannot guarantee that the model accurately represents f_{i_k}. Hence, if $\delta_k > \beta\pi_k$, the radius δ_k is reduced in an attempt to find a more accurate model.

- In the case where there was a successful iteration ($\rho_k \geq \eta$) and another function is taken as the representative of f_{min} ($i_{k+1} \neq i_k$), what we call "index swap", we increase δ_{k+1} and Δ_{k+1} by a factor of $\tau_4 \geq 1$. After a successful iteration with $\rho_k \geq \eta_1$ and index swap, this procedure is executed at most $\Gamma_{\max} \in \mathbb{N}$ iterations with index swap that satisfy $\eta \leq \rho_k < \eta_1$. Thus, the radii of the other iterations satisfying $\rho_k < \eta_1$ continue to be reduced by τ_1. By allowing the radius of the trust-region to increase, we expect the algorithm to try larger steps when a new $f_{i_{k+1}}$ is considered. In [1], Δ_{k+1} is reset to Δ_0 whenever index swap occurs, but we observed that such choice resulted in worst-complexity results with higher order.

- In the case where $|\mathcal{I}| = 1$ (usual convex constrained optimization problem), Algorithm 1 is reduced to the derivative-free trust-region algorithm proposed by [72].

Remark 3.7. *Algorithm 1 can be modified to allow iterations satisfying $\rho \geq 0$ to enter the radii adjustment phase (line 14). This change does not affect the results obtained in Section 3.3.*

3.3 Convergence analysis

In this section, we will provide global convergence results, in the LOVO sense, for sequences generated by Algorithm 1. We will show that every point of accumulation of a sequence generated by Algorithm 1 is stationary for some index $i \in \mathcal{I}$.

In the following lemma, we prove that the Hessian of the model m_k is bounded.

Lemma 3.8. *Suppose that Assumptions 3.2 and 3.5 hold. Thus, for every iteration $k \in \mathbb{N}$,*

$$\|\mathbf{H}_k\| \leq \kappa_H - 1,$$

where $\kappa_H := 2\kappa_g + L + 1$ and $L := \max_{i \in \mathcal{I}}\{L_i\}$.

Proof. Let $k \in \mathbb{N}$, $i_k \in \mathcal{I}_{min}(x_k)$ be the chosen index associated with m_k in this iteration and $d \in \mathbb{R}^n$ an arbitrary direction satisfying $\|d\| = \delta_k$. Initially, note that,

$$\nabla \mathrm{m}_k(x_k + d) = \mathbf{g}_k + \mathbf{H}_k d \text{ and } \nabla \mathrm{m}_k(x_k) = \mathbf{g}_k. \tag{3.6}$$

and therefore,

$$\begin{aligned}
\|\mathbf{H}_k d\| &= \|\nabla \mathrm{m}_k(x_k + d) - \nabla \mathrm{m}_k(x_k)\| \\
&\leq \|\nabla \mathrm{m}_k(x_k + d) - \nabla f_{i_k}(x_k + d)\| + \|\nabla f_{i_k}(x_k + d) - \nabla f_{i_k}(x_k)\| \\
&\quad + \|\nabla f_{i_k}(x_k) - \nabla \mathrm{m}_k(x_k)\| \\
&\leq L_{i_k}\delta_k + \|\nabla \mathrm{m}_k(x_k + d) - \nabla f_{i_k}(x_k + d)\| \qquad (3.7) \\
&\quad + \|\nabla f_{i_k}(x_k) - \nabla \mathrm{m}_k(x_k)\| \\
&\leq L_{i_k}\delta_k + 2\kappa_g \delta_k \\
&= (L_{i_k} + 2\kappa_g)\delta_k.
\end{aligned}$$

So, by the definition of the Euclidean matrix norm, it follows that

$$\begin{aligned}
\|\mathbf{H}_k\| &= \max_{\|d\| = \delta_k} \left\| \mathbf{H}_k \frac{d}{\|d\|} \right\| \\
&= \frac{1}{\delta_k} \max_{\|d\| = \delta_k} \|\mathbf{H}_k d\| \\
&\leq \frac{1}{\delta_k}(L_{i_k} + 2\kappa_g)\delta_k \\
&= L_{i_k} + 2\kappa_g \\
&\leq L + 2\kappa_g.
\end{aligned}$$

Thus, by arbitrarily choosing $k \in \mathbb{N}$, we obtain the desired result. □

Analogously to [17, 72], let us consider the following sets of indices

$$\mathcal{S} = \{k \in \mathbb{N} \mid \rho_k \geq \eta\} \text{ and } \overline{\mathcal{S}} = \{k \in \mathbb{N} \mid \rho_k \geq \eta_1\}, \tag{3.8}$$

and note that $\overline{\mathcal{S}} \subset \mathcal{S}$. The next lemma establishes that if the trust-region radius is sufficiently small, then the iteration will be successful.

Lemma 3.9. *Suppose that Assumptions 3.2, 3.5 and 3.6 hold. Consider the set*

$$\mathcal{K} = \left\{ k \in \mathbb{N} : \Delta_k \leq \min \left\{ \frac{\pi_k}{\kappa_H}, \frac{(1-\eta_1)\pi_k}{c_1}, \beta\pi_k, 1 \right\} \right\}, \tag{3.9}$$

where $c_1 := \big(L + \kappa_g + \kappa_H/2\big)/\theta$. *If* $k \in \mathcal{K}$, *then* $k \in \overline{\mathcal{S}}$.

Proof. Let $k \in \mathcal{K}$ and consider $i_k \in \mathcal{I}_{min}(x_k)$ the index associated with the model m_k in this iteration. By the Mean Value Theorem, there exists $\xi_k \in (0,1)$ such that

$$f_{i_k}(x_k + d_k) = f_{i_k}(x_k) + \nabla f_{i_k}(x_k + \xi_k d_k)^T d_k.$$

Therefore, by Assumptions 3.2, 3.5 and Lemma 3.8, by following the same arguments as [17, Lemma 3.1], we obtain

$$|\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k) + f_{i_k}(x_k + d_k) - f_{i_k}(x_k)| \leq \left(L + \kappa_g + \frac{1}{2}\kappa_H \right) \Delta_k^2. \tag{3.10}$$

By (3.9) we have that $\pi_k > 0$. Thus, the sufficient decrease condition of Assumption 3.6 implies $\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k) > 0$. The key point in this proof is to observe that, since $i_k \in \mathcal{I}_{min}(x_k)$, we know that $f_{i_k}(x_k) = f_{min}(x_k)$ and $f_{min}(x_k + d_k) \leq f_{i_k}(x_k + d_k)$. By (3.5), (3.10), Assumption 3.6 and Lemma 3.8,

$$\begin{aligned}
1 - \rho_k &= 1 - \frac{f_{min}(x_k) - f_{min}(x_k + d_k)}{\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)} \\
&\leq 1 - \frac{f_{i_k}(x_k) - f_{i_k}(x_k + d_k)}{\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)} \\
&\leq \frac{|\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k) - f_{i_k}(x_k) + f_{i_k}(x_k + d_k)|}{\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)} \\
&\leq \frac{\big(L + \kappa_g + \frac{1}{2}\kappa_H\big)\Delta_k^2}{\theta\pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\}} \\
&= \frac{c_1\Delta_k^2}{\pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\}},
\end{aligned}$$

where $c_1 = \left(L + \kappa_g + \kappa_H/2\right)/\theta$. By the definition of the set $\mathcal{K}$, we have that

$$\Delta_k \leq \min\left\{ \frac{\pi_k}{\kappa_H}, \frac{(1-\eta_1)\pi_k}{c_1}, \beta\pi_k, 1 \right\},$$

and so $\Delta_k = \min\left\{\pi_k/\kappa_H, \Delta_k, 1\right\}$. Hence,

$$1 - \rho_k \leq \frac{c_1\Delta_k^2}{\pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\}} = \frac{c_1\Delta_k^2}{\pi_k\Delta_k} = \frac{c_1\Delta_k}{\pi_k} \leq 1 - \eta_1,$$

from which we conclude that $\eta_1 \leq \rho_k$, and $k \in \overline{\mathcal{S}}$. Consequently $\mathcal{K} \subseteq \overline{\mathcal{S}}$. □

The following result is a direct consequence of Lemma 3.9. Assuming that $\pi_k \geq \varepsilon > 0$, we can prove that the trust-region radius is bounded below by $\Delta_{\min}$, which depends on the parameters of the algorithm, the characteristics of problem (3.1) and the type of model m_k used.

Corollary 3.10. *Suppose that Assumptions 3.2, 3.5 and 3.6 hold, and let $\varepsilon > 0$. If $\pi_k \geq \varepsilon$, then*

$$\Delta_k \geq \Delta_{\min} := \min\left\{\Delta_0, \tau_1 \min\left\{\frac{\varepsilon}{\kappa_H}, \frac{(1-\eta_1)\varepsilon}{c_1}, \beta\varepsilon, 1\right\}\right\}.$$

As mentioned earlier, the radius of the sample region δ_k controls the quality of the model. More specifically, Assumption 3.5 tells us that the smaller the sample radius, the better the models represent f_{i_k}. Therefore, it is reasonable to expect δ_k to go to zero. The following two lemmas show us that this happens.

Lemma 3.11. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold, and $\left|\overline{\mathcal{S}}\right| = \infty$. Then the sequence $\{\delta_k\}_{k\in\overline{\mathcal{S}}}$ converges to zero.*

Proof. The proof follows the same arguments as [72, Lemma 3]. Initially, note that by the definition of the set $\overline{\mathcal{S}}$, we have by the statement of the lemma that infinite successful iterations occur satisfying $\rho_k \geq \eta_1$.

Let $k \in \overline{\mathcal{S}}$, be arbitrary. Given that k is a successful iteration, we know that $\delta_k \leq \beta\pi_k$, since the iteration has not entered the criticality phase, where $\beta > 0$ is a parameter of the algorithm, and by the mechanisms of updating and adjusting of the radii, we have that $\delta_k \leq \Delta_k$. Thus, for any $k \in \overline{\mathcal{S}}$, by (3.5), Assumption 3.6 and Lemma 3.8,

$$\begin{aligned} f_{min}(x_k) - f_{min}(x_k + d_k) &= \rho_k\big(\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)\big) \\ &\geq \rho_k\theta\pi_k \min\left\{\frac{\pi_k}{1 + \|\mathbf{H}_k\|}, \Delta_k, 1\right\} \\ &\geq \rho_k\theta\pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\} \\ &\geq \frac{\eta_1\theta}{\beta}\delta_k \min\left\{\frac{\delta_k}{\beta\kappa_H}, \delta_k, 1\right\}. \end{aligned} \tag{3.11}$$

Since $\{f_{min}(x_k)\}_{k\in\mathbb{N}}$ is a nonincreasing monotone sequence and bounded below, because by Assumption 3.3, the funtions f_i, $i = 1, \ldots, r$, are bounded below in Ω, the left-hand side of the expression (3.11) converges to zero. Therefore, we can conclude that $\lim_{k\in\overline{\mathcal{S}}} \delta_k = 0$. □

Lemma 3.12. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold. Then the sequence* $\{\delta_k\}_{k\in\mathbb{N}}$ *converges to zero.*

Proof. The proof is based on [72, Lemma 3], but we will reproduce it for completeness. We will split the demonstration into two cases.

i) $\overline{\mathcal{S}}$ is finite.

Initially consider that the set $\overline{\mathcal{S}}$ is finite. Hence, there is a finite number of iterations for which the radii adjustment phase (line 14) is called. In fact, such phase is called for at most $\Gamma_{\max} \left|\overline{\mathcal{S}}\right|$ iterations. Thus, there is $k_0 \in \mathbb{N}$ such that for every $k \geq k_0$, the iteration $k \notin \overline{\mathcal{S}}$ and the radii adjustment phase (line 14) is not called. Let $\mathcal{M}_1 := \{k \in \mathbb{N} \mid k \geq k_0\}$. Note that, by the radii update phase (line 16) present in the algorithm, we have that for every iteration $k \in \mathcal{M}_1$ $\delta_{k+1} = \tau_1\delta_k$, where $\tau_1 < 1$. Therefore, given that δ_{k_0} is a constant, it follows that

$$\lim_{k\to\infty} \delta_k = \lim_{k\in\mathcal{M}_1} \delta_k = \lim_{k\to\infty} \tau_1^k \delta_{k_0} = 0.$$

ii) $\overline{\mathcal{S}}$ is infinite.

Now, assume that the set $\overline{\mathcal{S}}$ is infinite and let $\mathcal{M}_2 := \{k \in \mathbb{N} \mid k \notin \overline{\mathcal{S}}\}$. If the set $\mathcal{M}_2$ is finite, then by Lemma 3.11 it follows that

$$\lim_{k\to\infty} \delta_k = \lim_{k\in\overline{\mathcal{S}}} \delta_k = 0.$$

This leaves only the case where $\mathcal{M}_2$ is infinite. Note that, by Lemma 3.11, we obtain the desired limit for indices in $\overline{\mathcal{S}}$. We will extend this result to the entire sequence. For this purpose, let $k \in \mathcal{M}_2$ and l_k be the last iteration in $\overline{\mathcal{S}}$ before k. Since between the iterations l_k and k the radii adjustment phase (line 14) is called at most $\Gamma_{\max}$ times and $\tau_4 \geq \tau_3 \geq 1$, we have that the value of δ_k is at most $\tau_4^{\Gamma_{\max}}\delta_{l_k}$. Therefore,

$$\lim_{k\in\mathcal{M}_2} \delta_k \leq \lim_{k\in\mathcal{M}_2} \tau_4^{\Gamma_{\max}}\delta_{l_k} = \tau_4^{\Gamma_{\max}} \lim_{l_k\in\overline{\mathcal{S}}} \delta_{l_k} = 0,$$

and we conclude the proof. □

Next, we show a weak convergence result for the problem of minimizing the model m_k in the feasible region Ω.

Lemma 3.13. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold. Then sequence* $\{\pi_k\}_{k\in\mathbb{N}}$ *admits a subsequence that converges to zero.*

Proof. We will show that

$$\liminf_{k\to\infty} \pi_k = 0.$$

In fact, suppose by contradiction that there are $\varepsilon > 0$ and an integer $k_0 > 0$ such that $\pi_k \geq \varepsilon$ for all $k \geq k_0$. Let $k \in \overline{\mathcal{S}}$, with $k \geq k_0$ arbitrary. By the definition of ρ_k given in (3.5), Assumption 3.6, the contradiction hypothesis and Corollary 3.10, it follows that

$$\begin{aligned} f_{min}(x_k) - f_{min}(x_k + d_k) &\geq \rho_k \theta \pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\} \\ &\geq \eta_1 \theta \varepsilon \min\left\{\frac{\varepsilon}{\kappa_H}, \Delta_{\min}, 1\right\}. \end{aligned}$$

By Assumption 3.3, $\{f_{min}(x_k)\}_{k\in\mathbb{N}}$ is bounded below, and given that it is a monotone nonincreasing sequence, it follows that $f_{min}(x_k)-f_{min}(x_k+d_k) \to 0$. Since the right-hand side of the above inequality is a positive constant, the set $\{k \in \overline{\mathcal{S}} : k \geq k_0\}$ is finite, and thus the radii adjustment phase (line 14) is also called only a finite number of iterations. However, by the Lemma 3.12 we have that $\delta_k \to 0$, and given that $\pi_k \geq \varepsilon$ for all $k \geq k_1$, it follows that the criticality phase (line 3) is called only for a finite number of iterations. Thus, for every sufficiently large $k \in \mathbb{N}$, we only have iterations with $\rho_k < \eta_1$ without the radii adjustment phase (line 14) being called, which implies that $\Delta_{k+1} = \tau_1\Delta_k$. Consequently, $\Delta_k \to 0$, contradicting Corollary 3.10. □

The following result is a direct consequence of Lemma 3.13.

Corollary 3.14. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold. Then there is an index $i \in \mathcal{I}$, that is selected an infinite number of iterations such that*

$$\liminf_{k \in \mathcal{J}(i)} \pi_k = 0,$$

where $\mathcal{J}(i) = \{k \in \mathbb{N} \mid i_k = i\}$.

Proof. From Lemma 3.13, we know that $\{\pi_k\}_{k\in\mathbb{N}}$ admits a subsequence that converges to zero. Given that such a subsequence is infinite and that $|\mathcal{I}| = r < \infty$, there must be an index $i \in \mathcal{I}$ that is chosen in an infinite number of iterations of that subsequence. Thus, collecting such iterations into the set $\mathcal{J}(i) = \{k \in \mathbb{N} \mid i_k = i\}$, we have that $\lim\limits_{k\in\mathcal{J}(i)} \pi_k = 0$, and we conclude the proof. $\square$

If we, in addition, ask for a sufficient decrease condition $\eta > 0$ in Algorithm 1, then we can show that $\{\pi_k\}_{k\in\mathbb{N}}$ converges to zero for any subsequence when an index $i \in \mathcal{I}$ is selected an infinite number of iterations.

Lemma 3.15. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold, and $\eta > 0$. Let $i \in \mathcal{I}$ be an index chosen an infinite number of iterations. Thus,*

$$\lim_{k \in \mathcal{J}(i)} \pi_k = 0,$$

where $\mathcal{J}(i) = \{k \in \mathbb{N} \mid i_k = i\}$.

Proof. Initially, we know that $\mathcal{J}(i)$ exists by the same arguments of Corollary 3.14. Suppose by contradiction that for some $\varepsilon > 0$ the set $\mathcal{M}_1 = \mathcal{J} \cap \{k \in \mathbb{N} : \pi_k \geq \varepsilon\}$ is infinite. Given $k \in \mathcal{M}_1$, consider $u_k \in \mathbb{N}$ the first iteration such that $u_k > k$ and $\pi_{u_k} \leq \varepsilon/2$, and so $\pi_k - \pi_{u_k} \geq \varepsilon/2$. Note that the existence of u_k is guaranteed by Lemma 3.13. By Lemma 3.12, there is $k_0 \in \mathbb{N}$ such that, for $k \geq k_0$, we have $\delta_k \leq \varepsilon/(8\kappa_g)$, where κ_g is the constant given in Assumption 3.5. Also, let us define $\mathcal{C}_k = \{j \in \mathcal{S} : k \leq j < u_k\}$. We will show that $\mathcal{C}_k$ is nonempty by considering two cases: $u_k \in \mathcal{J}(i)$ and $u_k \notin \mathcal{J}(i)$.

i) $u_k \in \mathcal{J}(i)$. Let $k \in \mathcal{M}_1$, with $k \geq k_0$, and assume that $u_k \in \mathcal{J}(i)$. Using the definition of π_k, the triangular inequality, and the linearity and contraction properties of the projection operator, we have

$$\begin{aligned}\frac{\varepsilon}{2} &\le \pi_k - \pi_{u_k} \\ &= \|P_\Omega(x_k - \mathbf{g}_k) - x_k\| - \left\|P_\Omega(x_{u_k} - \mathbf{g}_{u_k}) - x_{u_k}\right\| \\ &\le \left\|P_\Omega(x_k - \mathbf{g}_k) - x_k - P_\Omega(x_{u_k} - \mathbf{g}_{u_k}) + x_{u_k}\right\| \\ &\le 2\,\|x_k - x_{u_k}\| + \left\|\mathbf{g}_k - \mathbf{g}_{u_k}\right\|.\end{aligned} \tag{3.12}$$

Given that $u_k \in \mathcal{J}(i)$, we have that $i_k = i_{u_k}$. So, by the triangular inequality and Assumptions 3.2 and 3.5, we obtain

$$\begin{aligned}\frac{\varepsilon}{2} &\le 2\,\|x_k - x_{u_k}\| + \left\|\mathbf{g}_k - \mathbf{g}_{u_k}\right\| \\ &\le \|\mathbf{g}_k - \nabla f_{i_k}(x_k)\| + \left\|\nabla f_{i_k}(x_k) - \nabla f_{i_{u_k}}(x_{u_k})\right\| + \left\|\nabla f_{i_{u_k}}(x_{u_k}) - \mathbf{g}_{u_k}\right\| \\ &\qquad + 2\,\|x_k - x_{u_k}\| \\ &\le 2\,\|x_k - x_{u_k}\| + \left\|\nabla f_{i_k}(x_k) - \nabla f_{i_{u_k}}(x_{u_k})\right\| + \kappa_g(\delta_k + \delta_{u_k}) \\ &\le (2 + L)\,\|x_k - x_{u_k}\| + \kappa_g(\delta_k + \delta_{u_k}) \\ &\le (2 + L)\,\|x_k - x_{u_k}\| + \frac{\varepsilon}{4},\end{aligned} \tag{3.13}$$

where the last inequality comes from the choice of k_0. Therefore, it follows that

$$\|x_k - x_{u_k}\| \ge \frac{\varepsilon}{4(2 + L)}. \tag{3.14}$$

Since $\varepsilon > 0$, if $\mathcal{C}_k = \emptyset$ then $x_k = x_{u_k}$, which contradicts (3.14). Hence $\mathcal{C}_k$ is nonempty.

ii) $u_k \notin \mathcal{J}(i)$. In this case, there is no guarantee that ∇f_{i_k} and $\nabla f_{i_{u_k}}$ are the same functions and, therefore, the argument used in item i) is not valid. However, given that $u_k \notin \mathcal{J}(i)$, by the index choice condition in the step acceptance phase (line 8), we know that there was at least one index swap between iterations k and u_k. Thus, the set $\mathcal{C}_k$ is nonempty.

Therefore, in both cases we obtain that $\mathcal{C}_k$ is a nonempty set. Thus, by Assumption 3.6, Lemma 3.8, Corollary 3.10, and the fact that $\pi_j \ge \varepsilon/2$, for all $j \in \mathcal{C}_k$, we have that, for all $k \in \mathcal{M}_1$, $k \ge k_0$,

$$
\begin{aligned}
f_{min}(x_k) - f_{min}(x_{u_k}) &\geq \sum_{j \in \mathcal{C}_k} \big(f_{min}(x_j) - f_{min}(x_{j+1})\big) \\
&\geq \sum_{j \in \mathcal{C}_k} \rho_j \theta \pi_j \min\left\{\frac{\pi_j}{\kappa_H}, \Delta_j, 1\right\} \\
&\geq \frac{\eta\theta\varepsilon}{2} \min\left\{\frac{\varepsilon}{2\kappa_H}, \sum_{j \in \mathcal{C}_k} \Delta_j, 1\right\} \\
&\geq \frac{\eta\theta\varepsilon}{2} \min\left\{\frac{\varepsilon}{2\kappa_H}, \Delta_{\min}, 1\right\}.
\end{aligned}
\tag{3.15}
$$

Thus, since $\eta > 0$, it follows that the right-hand side of (3.15) is a positive constant. On the other hand, by Assumption 3.3, $\{f_{min}(x_k)\}_{k \in \mathbb{N}}$ is bounded below and, by the construction of the algorithm, is a monotone nonincreasing sequence. Hence, $f_{min}(x_k) - f_{min}(x_{u_k}) \to 0$, which is a contradiction with (3.15), and completes the proof. □

We are now ready to prove the global convergence results for Algorithm 1. In a way similar to the stationarity measure π_k when solving subproblems (3.4), we adopt the criticality measure $\|\mathcal{P}_\Omega(x - \nabla f(x)) - x\|$, proposed by Conn, Gould and Toint [20] for optimization problems involving a continuous function f on a convex feasible set Ω. This measure was used in [17, 21, 72], for example, to establish global convergence results. The following theorem is the main result of this section. It allows us to further describe what kind of stationarity can be achieved by Algorithm 1.

Theorem 3.16. *Let us define sets $\mathcal{J}(i) = \{k \in \mathbb{N} \mid i_k = i\}$, for $i \in \mathcal{I}$. Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold and let $\{x_k\}_{k \in \mathbb{N}}$ be a sequence generated by Algorithm 1. The following statements are valid.*

i) If $\eta = 0$, then

$$\liminf_{k \to \infty} \|\mathcal{P}_\Omega(x_k - \nabla f_{i_k}(x_k)) - x_k\| = 0.$$

Furthermore, there is an index $i \in \mathcal{I}$ such that

$$\liminf_{k \in \mathcal{J}(i)} \|\mathcal{P}_\Omega(x_k - \nabla f_{i_k}(x_k)) - x_k\| = 0.$$

ii) If $\eta > 0$ and $i \in \mathcal{I}$ is any index chosen for an infinite number of iterations, then

$$\lim_{k \in \mathcal{J}(i)} \|\mathcal{P}_\Omega(x_k - \nabla f_{i_k}(x_k)) - x_k\| = 0.$$

iii) If $i \in \mathcal{I}$ is an index chosen for an infinite number of iterations and $x_* \in \mathbb{R}^n$ is an accumulation point for the subsequence $\{x_k\}_{k \in \mathcal{J}(i)} \subseteq \{x_k\}_{k \in \mathbb{N}}$ then $i \in \mathcal{I}_{min}(x_*)$.

Proof. Initially, note that by the triangular inequality, the properties of the projection operator, and Assumption 3.5, we have

$$\begin{aligned}
&\|P_\Omega(x_k - \nabla f_{i_k}(x_k)) - x_k\| \\
&\quad = \|P_\Omega(x_k - \nabla f_{i_k}(x_k)) - P_\Omega(x_k - \mathbf{g}_k) + P_\Omega(x_k - \mathbf{g}_k) - x_k\| \\
&\quad \leq \|P_\Omega(x_k - \nabla f_{i_k}(x_k)) - P_\Omega(x_k - \mathbf{g}_k)\| + \|P_\Omega(x_k - \mathbf{g}_k) - x_k\| \\
&\quad = \|P_\Omega(x_k - \nabla f_{i_k}(x_k) - x_k + \mathbf{g}_k)\| + \|P_\Omega(x_k - \mathbf{g}_k) - x_k\| \\
&\quad = \|P_\Omega(\mathbf{g}_k - \nabla f_{i_k}(x_k))\| + \|P_\Omega(x_k - \mathbf{g}_k) - x_k\| \\
&\quad \leq \|\mathbf{g}_k - \nabla f_{i_k}(x_k)\| + \|P_\Omega(x_k - \mathbf{g}_k) - x_k\| \\
&\quad \leq \kappa_g \delta_k + \pi_k.
\end{aligned} \tag{3.16}$$

Thus, using Lemmas 3.12 and 3.13, Corollary 3.14, and (3.16), we obtain i). On the other hand, by Lemmas 3.12 and 3.15, and expression (3.16), we conclude the proof of ii).

It still remains for us to prove iii). Note that by the statement of the theorem we have that $\lim_{k \in \mathcal{J}(i)} x_k = x_*$, and by Assumption 3.2, the functions f_j, $j \in \mathcal{I}$, are continuously differentiable. So the continuity of f_j gives us

$$\lim_{k \in \mathcal{J}(i)} f_j(x_k) = f_j(x_*), \tag{3.17}$$

for any index $j \in \mathcal{I}$. Given that $i \in \mathcal{I}_{min}(x_k)$, for every $k \in \mathcal{J}(i)$, we have by the definition of this set that $f_i(x_k) \leq f_j(x_k)$ for any index $j \in \mathcal{I}$ and $k \in \mathcal{J}(i)$. Thus, taking the limit in $\mathcal{J}(i)$, by (3.17) it follows that $f_i(x_*) \leq f_j(x_*)$, and therefore $i \in \mathcal{I}_{min}(x_*)$. □

Theorem 3.16 states that if $\eta > 0$ and $x_* \in \mathbb{R}^n$ is an accumulation point of a sequence $\{x_k\}_{k \in \mathbb{N}}$ generated by Algorithm 1, it is possible to construct a subsequence $\{x_k\}_{k \in \mathcal{J}(i)}$ that converges to x_*, where $\mathcal{J}(i) = \{k \in \mathbb{N} \mid i_k = i\}$, and x^* satisfies a necessary optimality condition of gradient projected type for the problem

$$\begin{array}{ll} \text{minimize} & f_i(x) \\ \text{subject to} & x \in \Omega. \end{array} \tag{3.18}$$

In other words, x^* is a first-order stationary point for the problem (3.18) (see [20] and [25, p. 450]).

The following definition and corollaries help us to understand Theorem 3.16 from the perspective of the LOVO theory [2].

Definition 3.17. [2, p. 05] Given $x_* \in \Omega$ and established a necessary optimality condition for the problem (3.18), we say that

i) x_* is strongly critical if it satisfies a necessary optimality condition for (3.18), for all $i \in I_{min}(x_*)$.

ii) x_* is weakly critical if it satisfies a necessary optimality condition for (3.18), for some index $i \in I_{min}(x_*)$.

Corollary 3.18. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold, and $\eta > 0$. If $x_* \in \mathbb{R}^n$ is an accumulation point of a sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by Algorithm 1, then x_* is weakly critical.*

Proof. Let x_* be an accumulation point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by Algorithm 1. Thus, there is a subsequence $\{x_k\}_{k\in\mathcal{N}} \subseteq \{x_k\}_{k\in\mathbb{N}}$ that converges to x_*. If $i \in \mathcal{I}$ is an index that repeats for an infinite number of iterations in this subsequence then, as subsequence $\{x_k\}_{k\in\mathcal{J}(i)} \subseteq \{x_k\}_{k\in\mathcal{N}} \subseteq \{x_k\}_{k\in\mathbb{N}}$ converges to x_*, by iii) of Theorem 3.16 it follows that $i \in \mathcal{I}_{min}(x_*)$. Furthermore, from ii) of the same theorem, it follows that

$$\lim_{k\in\mathcal{J}(i)} \|\mathcal{P}_\Omega(x_k - \nabla f_i(x_k)) - x_k\| = 0,$$

and thus x_* satisfies a necessary optimality condition for problem (3.18). Therefore, x_* is a weakly critical point. □

Corollary 3.19. *Suppose that Assumptions 3.2, 3.3, 3.5 and 3.6 hold. If Algorithm 1 generates a sequence $\{x_k\}_{k\in\mathbb{N}}$ that converges to $x_* \in \mathbb{R}^n$, then x_* is weakly critical.*

Proof. Let $\{x_k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 1 and suppose that it converges to a point $x_* \in \mathbb{R}^n$. If

- $\eta = 0$, let $i \in \mathcal{I}$ be the index from statement i) of Theorem 3.16;
- $\eta > 0$, let $i \in \mathcal{I}$ be any index chosen for an infinite number of iterations;

Note that $\{x_k\}_{k \in \mathcal{J}(i)} \subseteq \{x_k\}_{k \in \mathbb{N}}$ converges to x_* and so, by iii) of Theorem 3.16, it follows that $i \in \mathcal{I}_{min}(x_*)$. Furthermore, by i) or ii) of the same theorem, it follows that $\lim_{k \in \mathcal{J}(i)} \|\mathcal{P}_\Omega(x_k - \nabla f_i(x_k)) - x_k\| = 0$ and thus x_* satisfies a necessary optimality condition for the problem (3.18). Therefore, x_* is a weakly critical point. $\square$

Theorem 3.16 highlights another very desirable feature of Algorithm 1. It can be used as the inner solver for general-constrained derivative-free algorithms, especially of the Inexact Restoration type [12, 34], a case where Ω is composed by linear constraints. In such a case, Algorithm 1 is able to generate sequences to *Approximate Gradient Projection* (AGP) points. The AGP condition is a strong practical necessary optimality condition [3], which is satisfied by every local minimizer of a nonlinear optimization problem without any constraint qualification.

Consider our initial problem (3.1), and note that the feasible set $\Omega \subset \mathbb{R}^n$ is convex, closed, and nonempty. Assume that Ω can be expressed by a set of expressions of the form $g_j \leq 0$, $j = 1, 2, \ldots, k_1$, and $h_l = 0$, $l = 1, 2, \ldots, k_2$, where the functions g_j are convex, and h_l are affine. In this context, our algorithm is able to generate sequences that satisfy a *Convex Approximate Gradient Projection* (C-AGP) condition [3, p. 635]. C-AGP can be more appropriate in some contexts and given a feasible point x_* that satisfies C-AGP and the *Mangasarian-Fromovitz* (MFCQ) constraint qualification then x_* also satisfies the KKT conditions for problem (3.18) [3, Theorem 3.2]. However, convergence to C-AGP points does not imply AGP points and vice versa. Now consider the situation where the functions g_j, $j = 1, 2, \ldots, k_1$, are linear. This case is particularly interesting when we have bound (or linear in general) constraints on the feasible set. Under these circumstances, we can employ a *Linear Approximate Gradient Projection* (L-AGP) criterion [3, p. 638], which is a first-order optimality condition stronger than the original AGP condition, in the sense that L-AGP implies AGP.

The last discussion in this section is related to Assumption 3.5. It is well known that such assumption is satisfied if, for example, linear or quadratic interpolation models are constructed using the so-called Λ-poised sample points [28], $\Lambda > 0$. Models with a relaxed interpolation condition were also considered in [67, 72], and are covered in detail in Chapter 2. Unfortunately, usually the techniques used to ensure well poisedness deal only with the unconstrained case. Powell [59] uses a technique to build good underdetermined quadratic interpolation models for box constraints. Hough and Roberts [42] present a weaker definition of "fully linear models" and extend the concept of Λ-poisedness. Unfortunately, this definition depends on a different stationarity measure, which requires a convex-constrained optimization problem to be solved in order to be verified. On the other hand, the projected gradient measure used by Algorithm 1 and by Theorem 3.16 has a closed form in many special cases, when Ω is a box or a hyper-sphere, for example. Another possibility to build well poised models using unconstrained strategies is to allow f_{min} to be evaluated in points outside Ω. That was the case in [18, 34]. When f_{min} is not defined outside Ω (also known as *hard constraints*), then the criticality measure and strategy from [42] can be adapted to Algorithm 1 and benefit from using two different radii.

3.4 Worst-case complexity

In this section we will perform the worst-case complexity analysis for Algorithm 1. For this purpose, let us first define the stationarity measure $\pi_k^f = \|\mathcal{P}_\Omega(x_k - \nabla f_{i_k}(x_k)) - x_k\|$. Given $\epsilon > 0$, we will bound the number of iterations necessary to verify $\pi_k^f < \epsilon$ in terms of the initial point, problem's and algorithm's constants, and ϵ. Note that, by Theorem 3.16, this condition can be satisfied at least for a subsequence generated by Algorithm 1.

Hence, let $k_\epsilon \in \mathbb{N}$ be the first iteration that $\pi_{k_\epsilon}^f < \epsilon$ is verified. In order to help counting the number of iterations up to k_ϵ, we define the sets

- $\mathcal{C}_\epsilon$, iterations $k \leq k_\epsilon$ that entered the criticality phase (line 3);
- $\mathcal{U}_\epsilon$, all unsuccessful iterations ($\rho_k < \eta$);
- $\mathcal{A}_\epsilon^{\mathcal{R}}$ and $\mathcal{A}_\epsilon^{\mathcal{NR}}$, all acceptable iterations ($\eta \leq \rho_k < \eta_1$) in which the radii adjustment phase (line 14) is and is not called, respectively;

- $\overline{\mathcal{S}}_\epsilon^{\mathcal{R}}$ and $\overline{\mathcal{S}}_\epsilon^{\mathcal{NR}}$, all successful iterations ($\rho_k \geq \eta_1$) in which the radii adjustment phase (line 14) is and is not called, respectively;

For the purposes of notation, we also define

- $\overline{\mathcal{S}}_\epsilon = \overline{\mathcal{S}}_\epsilon^{\mathcal{R}} \cup \overline{\mathcal{S}}_\epsilon^{\mathcal{NR}}$, the set of all successful iterations;
- $\mathcal{R}_\epsilon = \mathcal{U}_\epsilon \cup \mathcal{A}_\epsilon^{\mathcal{NR}}$, the set of all iterations in which there was necessarily a reduction in the trust-region radius;
- $\mathcal{N}_\epsilon = \mathcal{C}_\epsilon \cup \mathcal{U}_\epsilon \cup \mathcal{A}_\epsilon^{\mathcal{R}} \cup \mathcal{A}_\epsilon^{\mathcal{NR}} = \mathcal{C}_\epsilon \cup \mathcal{R}_\epsilon \cup \mathcal{A}_\epsilon^{\mathcal{R}}$, the set of all iterations that are not successful.

The following two results help us to establish a relation between the stationarity measures π_k and π_k^f.

Corollary 3.20. *Suppose that Assumption 3.5 holds. Then the stationarity measures π_k and π_k^f satisfy $\left|\pi_k - \pi_k^f\right| \leq \kappa_g \delta_k$.*

Proof. The result follows from the proof of Theorem 3.16. □

Lemma 3.21. *Suppose that Assumption 3.5 holds and that the criticality phase (line 3) is not called in iteration $k \in \mathbb{N}$. If $\pi_k^f \geq \epsilon$, then $\pi_k \geq c_2 \epsilon$, where $c_2 := 1/(1+\kappa_g \beta)$.*

Proof. Since $k \in \mathbb{N}$ is not a criticality iteration, we have that $\delta_k \leq \beta \pi_k$. Thus, by Assumption 3.5 and Lemma 3.20, it follows that

$$\epsilon \leq \pi_k^f \leq \left|\pi_k^f - \pi_k\right| + \pi_k \leq \kappa_g \delta_k + \pi_k \leq \kappa_g \beta \pi_k + \pi_k \leq (\kappa_g \beta + 1)\, \pi_k.$$

Therefore, by letting $c_2 = 1/(1+\kappa_g\beta)$, we obtain $\pi_k \geq c_2 \epsilon$. □

We are now able to count the number of successful iterations.

Theorem 3.22. *Suppose that Assumptions 3.2, 3.5, and 3.6 hold. Then Algorithm 1 needs a maximum of*

$$\left|\overline{\mathcal{S}}_\epsilon\right| \leq \frac{f_{min}(x_0) - M}{\theta \eta_1 c_2} \Delta_{\min}^{-1} \epsilon^{-1},$$

successful iterations to reach $\pi_{k_\epsilon}^f < \epsilon$, where $M := \min_{i \in \mathcal{I}}\{M_i\}$, and $\Delta_{\min}$ is defined in Corollary 3.10.

Proof. Let $k \in \overline{\mathcal{S}}_\epsilon$. Thus, by the definition of ρ_k, Lemma 3.21, and letting $\varepsilon = c_2\epsilon$ in Corollary 3.10, we have

$$\begin{aligned} f_{min}(x_k) - f_{min}(x_{k+1}) &= \rho_k \left(\mathrm{m}_k(x_k) - \mathrm{m}_k(x_{k+1})\right) \\ &\geq \rho_k \theta \pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\} \\ &\geq \eta_1 \theta \pi_k \min\left\{\frac{\pi_k}{\kappa_H}, \Delta_k, 1\right\} \\ &\geq \eta_1 \theta c_2 \epsilon \min\left\{\frac{c_2\epsilon}{\kappa_H}, \Delta_k, 1\right\} \\ &\geq \eta_1 \theta c_2 \epsilon \min\left\{\frac{c_2\epsilon}{\kappa_H}, \Delta_{\min}, 1\right\} \\ &= \theta \eta_1 c_2 \Delta_{\min} \epsilon, \end{aligned} \tag{3.19}$$

where the last inequality comes from the definition of $\Delta_{\min}$. Thus, by adding up (3.19) to every $k \in \overline{\mathcal{S}}_\epsilon$, we get

$$\begin{aligned} f_{min}(x_0) - f_{min}(x_{k_\epsilon}) &\geq \sum_{k \in \overline{\mathcal{S}}_\epsilon} \left(f_{min}(x_k) - f_{min}(x_{k+1})\right) \\ &\geq \sum_{k \in \overline{\mathcal{S}}_\epsilon} \theta \eta_1 c_2 \Delta_{\min} \epsilon \\ &= \theta \eta_1 c_2 \Delta_{\min} \epsilon \left|\overline{\mathcal{S}}_\epsilon\right|. \end{aligned}$$

Given that $f_{min}(x_{k_\epsilon}) \geq M$, then

$$f_{min}(x_0) - M \geq f_{min}(x_0) - f_{min}(x_{k_\epsilon}) \geq \theta \eta_1 c_2 \Delta_{\min} \epsilon \left|\overline{\mathcal{S}}_\epsilon\right|,$$

and it follows that,

$$\left|\overline{\mathcal{S}}_\epsilon\right| \leq \frac{f_{min}(x_0) - M}{\theta \eta_1 c_2} \Delta_{\min}^{-1} \epsilon^{-1}.$$

□

Next, we set an upper bound on the number of iterations that are not successful. Recall that the set of all iterations up to k_ϵ is given by $\overline{\mathcal{S}}_\epsilon \cup \mathcal{N}_\epsilon$.

Theorem 3.23. *Under the conditions established in Theorem 3.22, Algorithm 1 needs at most*

$$|\mathcal{N}_\epsilon| \leq \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|} + c_3 \left|\overline{\mathcal{S}}_\epsilon\right| \tag{3.20}$$

not successful iterations to reach $\pi^f_{k_\epsilon} < \epsilon$, *where* $c_3 = \Gamma_{\max} + \left(\Gamma_{\max} + 1\right)\frac{\log(\tau_4)}{\left|\log\left(\tau_2\right)\right|}$.

Proof. Initially, note that, by the description of Algorithm 1

- $\Delta_{k+1} \leq \tau_2 \Delta_k$, for all $k \in \mathcal{C}_\epsilon$;
- $\Delta_{k+1} = \tau_1 \Delta_k$, for all $k \in \mathcal{R}_\epsilon$;
- $\Delta_{k+1} \leq \tau_3 \Delta_k$, for all $k \in \overline{\mathcal{S}}_\epsilon^{\mathcal{NR}}$;
- $\Delta_{k+1} = \tau_4 \Delta_k$, for all $k \in \mathcal{A}_\epsilon^{\mathcal{R}} \cup \overline{\mathcal{S}}_\epsilon^{\mathcal{R}}$.

Thus, by applying Corollary 3.10 with $\varepsilon = c_2\epsilon$ and since $\tau_1 \leq \tau_2$ and $\tau_3 \leq \tau_4$, we must have

$$\begin{aligned}
\Delta_{\min} &\leq \Delta_{k_\epsilon} \\
&\leq \Delta_0 \cdot \tau_2^{|\mathcal{C}_\epsilon|} \cdot \tau_1^{|\mathcal{R}_\epsilon|} \cdot \tau_3^{\left|\overline{\mathcal{S}}_\epsilon^{\mathcal{NR}}\right|} \cdot \tau_4^{\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon^{\mathcal{R}}\right|} \\
&\leq \Delta_0 \cdot \tau_2^{|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon|} \cdot \tau_4^{\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon^{\mathcal{NR}}\right|} \\
&= \Delta_0 \cdot \tau_2^{|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon|} \cdot \tau_4^{\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|}.
\end{aligned}$$

Thereby,

$$\begin{aligned}
\log\left(\Delta_{\min}\right) &\leq \log\left(\Delta_0 \cdot \tau_2^{|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon|} \cdot \tau_4^{\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|}\right) \\
&= \left(|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon|\right)\log\left(\tau_2\right) + \left(\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|\right)\log\left(\tau_4\right) + \log\left(\Delta_0\right),
\end{aligned}$$

what provides us

$$\left(|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon|\right)\log\left(\tau_2\right) \geq \log\left(\Delta_{\min}\right) - \log\left(\Delta_0\right) - \left(\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|\right)\log\left(\tau_4\right).$$

Given that $\tau_2 < 1$, we have $\log\left(\tau_2\right) < 0$, it follows that

$$\begin{aligned}
|\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon| &\leq \frac{\log\left(\Delta_{\min}\right) - \log\left(\Delta_0\right) - \left(\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|\right)\log\left(\tau_4\right)}{\log\left(\tau_2\right)} \\
&= \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|} + \left(\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| + \left|\overline{\mathcal{S}}_\epsilon\right|\right)\frac{\log\left(\tau_4\right)}{\left|\log\left(\tau_2\right)\right|}.
\end{aligned}$$

By the radii adjustment phase (line 14), we know that after an iteration in $\overline{\mathcal{S}}_\epsilon$, at most $\Gamma_{\max}$ iterations of the type $\mathcal{A}_\epsilon^{\mathcal{R}}$ can occur, and so, $\left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| \leq \Gamma_{\max}\left|\overline{\mathcal{S}}_\epsilon\right|$. Therefore, it follows that

$$\begin{aligned}|\mathcal{N}_\epsilon| &= |\mathcal{C}_\epsilon| + |\mathcal{R}_\epsilon| + \left|\mathcal{A}_\epsilon^{\mathcal{R}}\right| \\ &\leq \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|} + \left(\Gamma_{\max} + 1\right)\left|\overline{\mathcal{S}}_\epsilon\right| \frac{\log\left(\tau_4\right)}{\left|\log\left(\tau_2\right)\right|} + \Gamma_{\max}\left|\overline{\mathcal{S}}_\epsilon\right| \\ &= \left(\Gamma_{\max} + \left(\Gamma_{\max} + 1\right)\frac{\log\left(\tau_4\right)}{\left|\log\left(\tau_2\right)\right|}\right)\left|\overline{\mathcal{S}}_\epsilon\right| + \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|},\end{aligned}$$

what, together with the definition of c_3 finishes the proof. □

Theorem 3.24. *Under the conditions of Theorem 3.22, Algorithm 1 needs at most $\mathcal{O}(\kappa_g^3\epsilon^{-2})$ iterations to reach $\pi_{k_\epsilon}^f < \epsilon$.*

Proof. By Theorems 3.22 and 3.23, we have

$$\begin{aligned}\left|\overline{\mathcal{S}}_\epsilon\right| + |\mathcal{N}_\epsilon| &\leq \left|\overline{\mathcal{S}}_\epsilon\right| + c_3\left|\overline{\mathcal{S}}_\epsilon\right| + \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|} \\ &= (1 + c_3)\left|\overline{\mathcal{S}}_\epsilon\right| + \frac{\log\left(\Delta_0\right) - \log\left(\Delta_{\min}\right)}{\left|\log\left(\tau_2\right)\right|} \\ &\leq \frac{(f_{min}(x_0) - M)}{\theta\eta_1 c_2} c_3 \Delta_{\min}^{-1}\epsilon^{-1} + \frac{\log\left(\Delta_0\right) + \log\left(\Delta_{\min}^{-1}\right)}{\left|\log\left(\tau_2\right)\right|}.\end{aligned} \tag{3.21}$$

From Lemma 3.9, we have that $c_1 = (L + \kappa_g + \kappa_H/2)\,\theta^{-1} = \mathcal{O}(\max\{\kappa_g, \kappa_H\})$. Similarly, by Lemma 3.21, $c_2^{-1} = 1 + \kappa_g\beta = \mathcal{O}(\kappa_g)$. Hence, by letting $\varepsilon = c_2\epsilon$ in Corollary 3.10,

$$\begin{aligned}\Delta_{\min}^{-1} &= \max\left\{\Delta_0^{-1}, \frac{\kappa_H}{\tau_1 c_2\epsilon}, \frac{c_1}{\tau_1(1-\eta_1)c_2\epsilon}, \frac{1}{\tau_1\beta c_2\epsilon}, \tau_1^{-1}\right\} \\ &= \mathcal{O}(\max\{\kappa_g, \kappa_H\}\,\kappa_g\epsilon^{-1}).\end{aligned} \tag{3.22}$$

Therefore, we get from (3.21), (3.22), and Lemma 3.9, that

$$\left|\overline{\mathcal{S}}_\epsilon\right| + |\mathcal{N}_\epsilon| = \mathcal{O}(\max\{\kappa_g, \kappa_H\}\,\kappa_g^2\epsilon^{-2}) = \mathcal{O}(\kappa_g^3\epsilon^{-2}),$$

since $\kappa_H = 2\kappa_g + L + 1$. □

Remark 3.25. *Similarly to [1, Algorithm 3.2], we can allow resets in the trust-region radius of type $\Delta_{k+1} = \max(\tau_3\Delta_k, \Delta_0)$ at line 15 in the radii adjustment phase. However, this choice worsens the upper bound presented in Theorem 3.24 to $\mathcal{O}(\epsilon^{-3})$.*

Remark 3.26. *Similarly to Garmanjani et al. [35], we chose not to include assumptions about the order of magnitude of the β parameter, which acts in accessing the criticality phase (line 3). Although it is desirable that β is taken in an inversely proportional way to the choice of ϵ, in the context of our algorithm that is not necessary to establish complexity results. Cartis and Roberts [13], when studying the worst-caes complexity of algorithm* `DFO-GN`*, need to assume a hypothesis about the magnitude of the criticality phase parameter.*

In the following result, we explicitly expand constant κ_g to show the worst-case complexity estimates for function evaluations. This is possible by further especifying how the models are constructed. We consider determined and underdetermined linear and quadratic polynomial models satisfying the inexact interpolation condition $|\mathrm{m}(y^j) - f(y^j)| \leq \kappa\delta^2$, for each point y^j in a Λ-poised sample set $\mathcal{Y} \subset \overline{B}(x_k, \delta_k)$, where $\kappa \geq 0$ is the inexact constant. Such calculations were given in Chapter 2. Note that this interpolation condition naturally includes the classical interpolation models when $\kappa = 0$. Comments on how to build and maintain Λ-poised sets were made in the end of Section 3.3 and practical considerations are subject to the next chapter.

Theorem 3.27. *Suppose that the Assumptions 3.2, 3.3, and 3.6 hold. Also, assume that the models* m *are constructed by inexact linear or quadratic interpolation using a Λ-poised set $\mathcal{Y} \subset \mathbb{R}^n$ of sample points, $\Lambda > 0$. Then, the number of function evaluations that Algorithm 1 needs to reach $\pi^f_{k_\epsilon} \leq \epsilon$ is*

i) $\mathcal{O}\big((n+r)n^3\epsilon^{-2}\big)$ *if the model is linear;*

ii) $\mathcal{O}\big((r+n^2)n^{12}\epsilon^{-2}\big)$ *if the model is quadratic determined;*

iii) $\mathcal{O}\big((r+p)n^{\frac{9}{2}}p^{\frac{15}{2}}\epsilon^{-2}\big)$ *if the model is quadratic underdetermined, for $n < p < (n^2+3n)/2$ is the number of points used.*

Proof. Initially, note that at each iteration, at most $|\mathcal{Y}|$ evaluations of the function f_i are performed, for some index $i \in \mathcal{I}$, in order to build a Λ-poised set from scratch. Moreover, a single evaluation of the f_{min} function is necessary to evaluate ρ_k, which in turn depends on $|\mathcal{I}| = r$ function evaluations. Thus, by Theorem 3.24, we will need at most $\mathcal{O}\big((|\mathcal{Y}|+r)\kappa_g^3\epsilon^{-2}\big)$ function evaluations during algorithm execution. We will separate the proof into three cases.

i) Linear case.

In this case $|\mathcal{Y}| = n+1$ and, by Theorem 2.5 and Lemma 2.7, we have that

$$\kappa_g = L + \left(\frac{L}{2} + 2\kappa\right)\Lambda n = \mathcal{O}(n).$$

Thus, we conclude that at most $\mathcal{O}\big((n+r)n^3\epsilon^{-2}\big)$ function evaluations are necessary.

ii) Quadratic case.

In this case $|\mathcal{Y}| = (n^2+3n)/2+1 = q+1$. Again, using Theorem 2.5 and Lemma 2.7, we obtain

$$\begin{aligned}
\kappa_g &= 2\Big(4\Lambda\sqrt{(q+1)^3}\Big)\sqrt{q}\big(1+\sqrt{2}\big)(\kappa+L)\\
&= 8\Lambda\big(1+\sqrt{2}\big)\sqrt{q}\sqrt{(q+1)^3}(\kappa+L)\\
&< 8\Lambda\big(1+\sqrt{2}\big)(q+1)^2(\kappa+L)\\
&= 8\Lambda\big(1+\sqrt{2}\big)\left(\frac{n^2+3n+2}{2}\right)^2(\kappa+L)\\
&= \mathcal{O}(n^4)
\end{aligned}$$

what gives to us at most $\mathcal{O}\big((r+n^2)n^{12}\epsilon^{-2}\big)$ function evaluations.

iii) Quadratic underdetermined case.

Finally, suppose that the models are quadratic underdetermined, with $|\mathcal{Y}| = p+1$ points, where $n < p < q$, q defined in case ii). Thus, by theorems 2.11, 2.19, and 2.22, it follows that

$$\begin{aligned}
\kappa_g &= 2\sqrt{p}\left(\Lambda\sqrt{2(n+1)}(p+1)\right)\left(L+\kappa+\left(\kappa+\frac{L}{2}\right)\frac{3\Lambda(p+1)\sqrt{2(q+1)}}{c(\delta_{\max})^2}\right)\\
&< 2\sqrt{2}\Lambda\sqrt{(n+1)}\sqrt{(p+1)^3}\left(L+\kappa+\left(\kappa+\frac{L}{2}\right)\frac{3\sqrt{2}\Lambda}{c(\delta_{\max})^2}(p+1)\sqrt{q+1}\right)\\
&= \mathcal{O}(n^{\frac{3}{2}}p^{\frac{5}{2}}).
\end{aligned}$$

Hence, we will need at most $\mathcal{O}\big((r+p)n^{\frac{9}{2}}p^{\frac{15}{2}}\epsilon^{-2}\big)$ function evaluations.

$\square$

It is also worth mentioning that in the context of derivative-free optimization, most of the worst-case complexity results presented in the literature are for direct-search methods of directional type based on a condition of sufficient decrease [35, p. 1988]. Among the works that study derivative-free trust-region methods with convergence to first-order stationary points, as Algorithm 1, we can highlight the bound $\mathcal{O}\big(\epsilon^{-2}\big)$ obtained by Garmanjani et al. [35] for unconstrained composite optimization problems, which is also obtained in expectation by Gratton et al. [39], but based on probabilistic models. Grapiglia et al. [37], on the other hand, presents the bound $\mathcal{O}\big(\,|\log(\epsilon)|\,\epsilon^{-2}\big)$ for unconstrained composite nonsmooth problems and problems with equality constraints. Unfortunately, we are not aware of worst-case complexity results for derivative-free trust-region methods for problems with general convex constraints.

Chapter 4

Numerical implementation and experiments

We dedicate this chapter to presenting our implementation of Algorithm 1, performing numerical tests using data profiles, and discussing the benchmark results. In Section 4.1, we present a brief overview of techniques for benchmarking derivative-free solvers. In particular, we address the techniques of performance profiles, data profiles and relative minimization profiles. In Section 4.2, we discuss the implementation details of our algorithm, which we will call `LOWDER`, an acronym for Low order-value Optimization Without DERivatives. Our chosen test sets and algorithms to benchmark `LOWDER` performance are described in Section 4.3, along with numerical results.

4.1 Short overview

In recent years, derivative-free optimization has become an increasingly popular research area, especially due to the large number of complex applications that can be solved [13]. The growing interest in the area also boosted a new wave of theoretical development and the proposition of algorithms for the most diverse classes of derivative-free optimization problems [52]. Recent surveys, such as those presented by Rios and Sahinidis [63] and Larson et al. [45], point to this growing trend. In particular, [63] lists at least 22 leading software implementations of state-of-the-art algorithms. On the other hand, [45] brings a detailed description of several classes of derivative-free optimization methods, as well as enumerates several algorithms.

Faced with this wide variety of derivative-free optimization methods and algorithms, a question that has been gaining visibility is whether there is any fair way of comparing. Although there are several works in this direction, such as [8, 22, 29, 30, 50, 52, 53], there is no consensus among researchers on the subject [52]. In addition, it is common to come across a huge amount of data when performing benchmark tests on large test sets, which can impair the visualization and synthesis of results, and discourage a more thorough analysis of the performance of algorithms.

Among the benchmark tools for derivative-free optimization algorithms, the most popular are perhaps performance profiles [30] and data profiles [52]. Performance profiles are developed to compare solvers using a performance ratio chosen by the user. In this sense, the performance profile of a solver is the cumulative distribution function for the performance metric [30]. Despite its popularity, this tool has some limitations, being often sensitive to the algorithm stopping criteria and the success criteria established by the user [9, 29]. In turn, data profiles manage to overcome most of the limitations presented, adding even more information since are capable of providing the percentage of problems solved for a given computational budget, considering a pre-established tolerance. However, [29] points out that the behavior of the data profile can also be influenced by the success criteria chosen by the user. Curtis et al. [29] also proposes a new technique called relative minimization profiles (RMP). According to the authors, this tool can assess simultaneously the relative performance of several algorithms, taking into account the objective function value, the viability of the point, and the speed of progress. Despite being a more robust comparison method than the options presented, RMP still has some sensitivity to user choices when dealing with tolerance in violation of feasibility [29, p. 166]. It is also worth reiterating that as this is a recent tool, the limitations of this technique still need to be studied.

For the numerical tests carried out in this chapter, we adopted data profiles as the benchmark tool, due to their wide use in the context of derivative-free optimization problems.

4.2 Implementation details

The `LOWDER` algorithm employs linear models to solve LOVO problems and has several practical improvements when compared to the theoretical algorithm presented in Section 3.2. One of the most significant changes occurs in definition of the relative reduction coefficient ρ_k. Note that we need to calculate ρ_k at each iteration in order to define the step acceptance, update, and radius correction phases. Since this involves evaluating the function f_{min}, which can be costly in computational terms, in practice, we allow `LOWDER` to employ the coefficient proposed by Castelani et al. [15]:

$$\hat{\rho}_k = \frac{f_{i_k}(x_k) - f_{i_k}(x_k + d_k)}{\mathrm{m}_k(x_k) - \mathrm{m}_k(x_k + d_k)},$$

where $i_k \in \mathcal{I}_{min}(x_k)$, for up to `nrhomax` $\in \mathbb{N}$ consecutive iterations. After `nrhomax` iterations ρ_k has to be calculated as presented in (3.5). Note that $\hat{\rho}_k \leq \rho_k$, and so we are being more demanding about the quality of the models and the decrease obtained. During the preliminary tests, `nrhomax` $= 3$ proved to be efficient.

Another relevant change is the way the solution $d_k \in \mathbb{R}^n$ of the trust region subproblem (3.4) is incorporated into the sample set $\mathcal{Y}$. To do so, we implemented simplified versions of the `TRSBOX` and `ALTMOV` functions, developed by Powell [59] for the `BOBYQA` algorithm. Initially, we compute a solution $d_k^{\mathtt{TRS}}$ using the `TRSBOX` algorithm for the linear case. If $\|d_k^{\mathtt{TRS}}\| \geq \frac{\Delta_k}{2}$ and $\rho_k > 0$ (or $\hat{\rho}_k > 0$), the point $x_{new} = x_k + d_k^{\mathtt{TRS}}$ is inserted into $\mathcal{Y}$. Otherwise, we run `ALTMOV` to look for an alternative direction $d_k^{\mathtt{ALT}}$, and add the point $x_{new} = x_k + d_k^{\mathtt{ALT}}$ to $\mathcal{Y}$. At each iteration of `LOWDER`, only one point is added to $\mathcal{Y}$, while another point is removed, following procedures similar to `BOBYQA`.

In this first version, `LOWDER` employs only linear models to solve problem (3.1). In this sense, `LOWDER` uses the sample set construction mechanism of `BOBYQA`. As with this solver, we avoided fully rebuilding the models due to the computational cost involved. In general, the model and sample set are only rebuilt if one of the following conditions is satisfied:

i) An index swap occurs, that is $i_{k+1} \neq i_k$, and the direction calculated by `TRSBOX` satisfies $\rho_k \geq \eta$ (or $\hat{\rho}_k \geq \eta$); or yet,

ii) An index swap occurs, $\rho_k \geq 0$ (or $\hat{\rho}_k \geq 0$) and $f_{min}(x_{new})$ is avaliable.

In all other cases, the model is updated using the current sample set and the information obtained about the point x_{new}. Recall that only evaluations of $f_{i_{k+1}}$ are needed to rebuild the model. Each time the model is rebuilt or updated, we also calculate the QR factorization of the matrix $\mathbf{L}_L$ defined in (2.3). This information is needed to calculate $d_k^{\texttt{ALT}}$ given by ALTMOV and the point that should leave set $\mathcal{Y}$ in case direction $d_k^{\texttt{TRS}}$ is accepted.

Given that the radius of the sample region δ_k controls the quality of the model, a solution $\tilde{x} = x_k \in \Omega$ is declared successful if it satisfies $\delta_k \leq \delta_{\min}$ and $\beta\pi_k \leq \delta_{\min}$ (or $\beta\tilde{\pi}_k \leq \delta_{\min}$), where $\delta_{\min} > 0$ is a parameter defined by the user. The default values for β and $\delta_{\min}$ are 1 and 10^{-8}, respectively. We say that `LOWDER` is stalled at iteration k if there is no more room for improvement in the sample set and the model. This situation translates into the case where $\delta_k \leq \delta_{\min}$, $\Delta_k \leq \delta_{\min}$ and it is not possible to perform any more `ALTMOV` iterations. In this case, the maximum number of consecutive iterations of the `ALTMOV` type is controlled by the `maxalt` $\in \mathbb{N}$ parameter, whose default value is $|\mathcal{Y}| - 1$. Note that if the sample and trust-region radii have values greater than $\delta_{\min}$, we allow more than `maxalt` consecutive iterations with `ALTMOV` directions. However, as soon as the stalled criterion is satisfied the algorithm exits. For safety, the algorithm also exits if more than `maxcrit` $\in \mathbb{N}$ successive iterations access the criticality phase (line 3), whose default value is $|\mathcal{Y}| - 1$.

The `LOWDER` solver was implemented in the `Julia` language, version 1.6.1, and is available in the repository:

`https://github.com/aschwertner/LOWDER`

4.3 Numerical experiments

To benchmark the `LOWDER`'s performance in solving LOVO black-box problems, we compared our solver with other algorithms able of solving derivative-free optimization problems. These algorithms are Manifold Sampling Primal (`MS-P`) and Nonlinear Optimization with the MADS algorithm (`NOMAD`).

The `MS-P` solver is a `MatLab` implementation of the Manifold Sampling method proposed in [44, 45], and is designed to solve bound-constrained nonsmooth composite minimization problems

$$\min_{x\in\Omega} f(x) = \min_{x\in\Omega} h\left(F(x)\right),$$

where $F : \mathbb{R}^n \to \mathbb{R}^p$, $h : \mathbb{R}^p \to \mathbb{R}$ is a continuous selection (see [44, Definition 1]), and the feasible set is a subset of the n-fold Cartesian product of the extended reals $\mathbb{R} \cup \{-\infty, \infty\}$ defined by bound constraints of the form $\Omega = \{x : l \leq x \leq u\}$. The manifold sampling algorithm was initially proposed by Larson et al. [45] as a variant of gradient sampling, and can be classified as model-based derivative-free method. In this method, models of F are combined with sampled information about the function h to construct local models called smooth master models, for use within a trust-region framework. `MS-P` builds fully linear quadratic models using the interpolation and regression mechanisms of `POUNDERS` [74]. We are grateful to Jeffrey Larson, co-author of the method, for providing the `MS-P` and `POUNDERS` codes and being available to help us.

The `NOMAD` software is a `C++` implementation of the Mesh Adaptive Direct Search (MADS) algorithm, solves black-box optimization problems in general, and uses the progressive barriers method to deal with problems with constraints. In this work, we use version 4.2.0 of `NOMAD` [6] through the interface for the `Julia` language called `NOMAD.jl`, version 2.2.1 [51]. This version of `NOMAD` presents a series of improvements compared to previous versions, especially in comparison to version 3.9, such as a new software architecture and support for new pool mechanisms and search algorithms, among others. All the algorithms were run with their default parameters on an AMD Ryzen 7 1700X 3.40GHz with 8 cores (16 threads) and 16GB of RAM and Linux Ubuntu Budgie 22.04.1 LTS operating system.

Our benchmark suite includes three sets of test problems denominated by MW, HS, and QD. The MW set contains problems proposed by Moré and Wild in [52] for benchmarking derivative-free optimization algorithms and comprises 53 unconstrained problems. These problems are variations of 22 nonlinear least squares functions taken from the `CUTEr` collection [36]. Each function is defined by r components in n variables. By combining different values for n and r, and distinct starting points, we obtain all the problems in MW test set.

The HS test set includes 87 bound-constrained problems selected and modified from the original collection published by Hock and Schittkowski [41] for testing nonlinear programming algorithms (see appendices A and B for a full description). Initially, we selected 8 problems from [41] with bound constraints and different objective functions. By combining two, three, or four of these problems, a new objective function f_{min} is defined. The problem dimension is the largest dimension among the combined problems, and the bound constraints are taken as the intersection of the original boxes. If this procedure generates constraints with fixed values, that is $l_i = u_i$ for some index $i \in \{1, \ldots, n\}$, then the problem is discarded.

Lastly, the QD test set has 5 subsets with 50 problems each, all with bound constraints. The purpose of this test set is to demonstrate the impact of the number of functions f_i of the objective function f_{min} in the performance of the algorithms. In this sense, the subsets of QD consist of problems with an increasing number of component functions based on the general formula:

$$f_i(x) = 5^i + \frac{1}{2}\sum_{j=1}^{n} a_j^i \cdot (x_j - b_j^i)^2,$$

where $a^i \in [0, 1000]^n$, and $b^i \in [0, 10]^n$ are vectors built by the pseudorandom number generator `MarsenneTwister` from the `Random` package of the `Julia` language, with the same seed. We generate the problems in the test subsets with $r \in \{10, 25, 50, 75, 100\}$ component functions, respectively. We employ the same seed to generate vectors a^i and b^i to ensure that the generated component functions come into problems belonging to test sets with larger r values. In this way, the component functions generated for the QD10 test subset are present in the other subsets, and they are complemented by new functions f_i as the value of r increases. In our case, all problems were generated with dimension $n = 10$, have the same bound constraints $l = [0, \ldots, 0]$ and $u = [10, \ldots, 10]$, and the starting point $x_0 \in \mathbb{R}^n$ is set to the center of the box. Figure 4.1 ilustrates a two dimensional example of objective function generated by this procedure, with 10 component functions.

We also present in Table 4.1 the maximum and minimum values for the dimension $n \in \mathbb{N}$ and the number of component functions $r \in \mathbb{N}$ of the problems of each test set.

We use data profiles [52] to compare the performance of `LOWDER`, `MS-P`, and `NOMAD`, over our benchmark test suite. We are interested in comparing the function values obtained by each of the algorithms. Thus, we consider that a method has solved a problem with

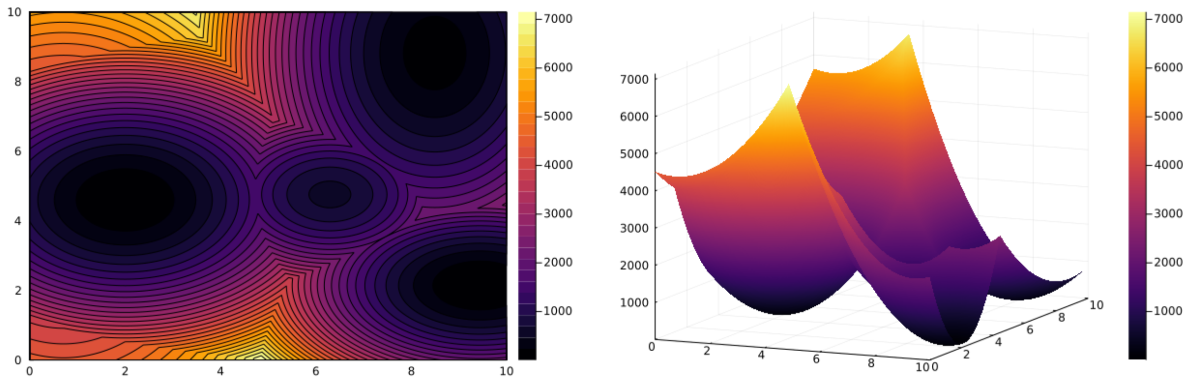


Figure 4.1: Contour plot (left) and surface plot (right) for an example of objective function f_{min} defined by the generating function of QD test set.

Test set	$n_{\min}$	$n_{\max}$	$r_{\min}$	$r_{\max}$
MW	2	12	2	65
HS	2	10	2	4
QD	10	10	10	100

Table 4.1: Dimension and number of component functions of MW, HS and QD test sets.

tolerance level $\tau > 0$ after t function evaluations if the iterate x^t satisfies

$$f(x^t) \leq f_L + \tau \left(f(x_0) - f_L \right), \tag{4.1}$$

where x_0 is the starting point of the problem common to all algorithms, and f_L is the smallest function value obtained by the solvers for a given budget of function evaluations. Following Moré and Wild's suggestion [52, Section 5], we decided to investigate the behavior of algorithms with a limit of 100 simplex gradients of the objective function, where one simplex gradient is defined by $n + 1$ function evaluations. Since `MS-P` and `NOMAD` always evaluate function f_{min} completely and `LOWDER` has the ability to run each component function f_i, $i \in \mathcal{I}$, independently, for each set of tests considered, we define the budget as $100\,(n_{max} + 1)$ evaluations of f_{min} for `MS-P` and `NOMAD`, and $100 r_p\,(n_{max} + 1)$ evaluations of f_i for `LOWDER`, where r_p is the number of component functions of the problem p. Therefore, we allow the algorithms to run at least 100 simplex gradients of the objective function f_{min} for each problem in the test set.

To build the data profiles, we recorded the function values of f_{min} accessed by the `MS-P` and `NOMAD` solvers and the function values of f_i computed by `LOWDER` for each one of the selected problems. Once the solver satisfies the criterion (4.1) for a problem p for the first time after t function evaluations, its data profile row is incremented on the vertical axis by $\frac{1}{|\mathcal{P}|}$ at the point $\frac{t}{(n_p+1)}$, in the case of `MS-P` and `NOMAD`, or at the point $\frac{t}{r_p(n_p+1)}$, for `LOWDER`. Therefore, our metric of simplex gradient evaluations for f_{min} is maintained.

The data profiles presented in the next subsections are available in a larger size in appendices C, D, and E. The codes used to generate the numerical tests and data profiles presented in this chapter are available at:

`https://github.com/aschwertner/LOWDER_Numerical_Tests`

4.3.1 MW test set

The MW set is our main problem test set since it was designed by Moré and Wild [52], especially to benchmark unconstrained derivative-free algorithms. In Figure 4.2, we present four data profiles for the MW test set. Each plot shows the percentage of problems solved for a specified tolerance τ as a function of a computational budget of simplex gradients of f_{min}. As we can see, `LOWDER` and `MS-P` solve around 90% of the problems for all tolerance levels τ, up to the fixed budget. The difference in robustness between this two solvers is less than 5%. We also note that the behavior of `NOMAD` its very different for distinct levels of tolerance, ranging from around 60% to less than 40%, with $\tau = 10^{-1}$ and $\tau = 10^{-7}$, respectively.

`LOWDER` is very efficient at solving problems in the MW test suite, solving about 90% of the problems with less than 20 simplex gradients. `MS-P` has a behavior very close to `LOWDER`, outperforming the latter by about 2% for $\tau = 10^{-1}$, but loses performance as the tolerance decreases, solving only 80% of problems for $\tau = 10^{-7}$. Although this behavior is expected, since `LOWDER` uses mainly information of the f_i's, we must recall that `MS-P` uses quadratic models, while `LOWDER` uses linear ones. For this computational budget and $\tau = 10^{-1}$, `NOMAD` can solve 50% of the problems in the test set. However its performance drops and stabilizes at around 40% for other values of tolerance. Inspecting the `NOMAD` execution data, we can see three distinct behaviors that can explain its low performance in the MW test set: difficulty in reducing the objective function, low convergence rate, and convergence to weak stationary points that are local minima, while `LOWDER` and `MS-P` can converge to global minimum points.

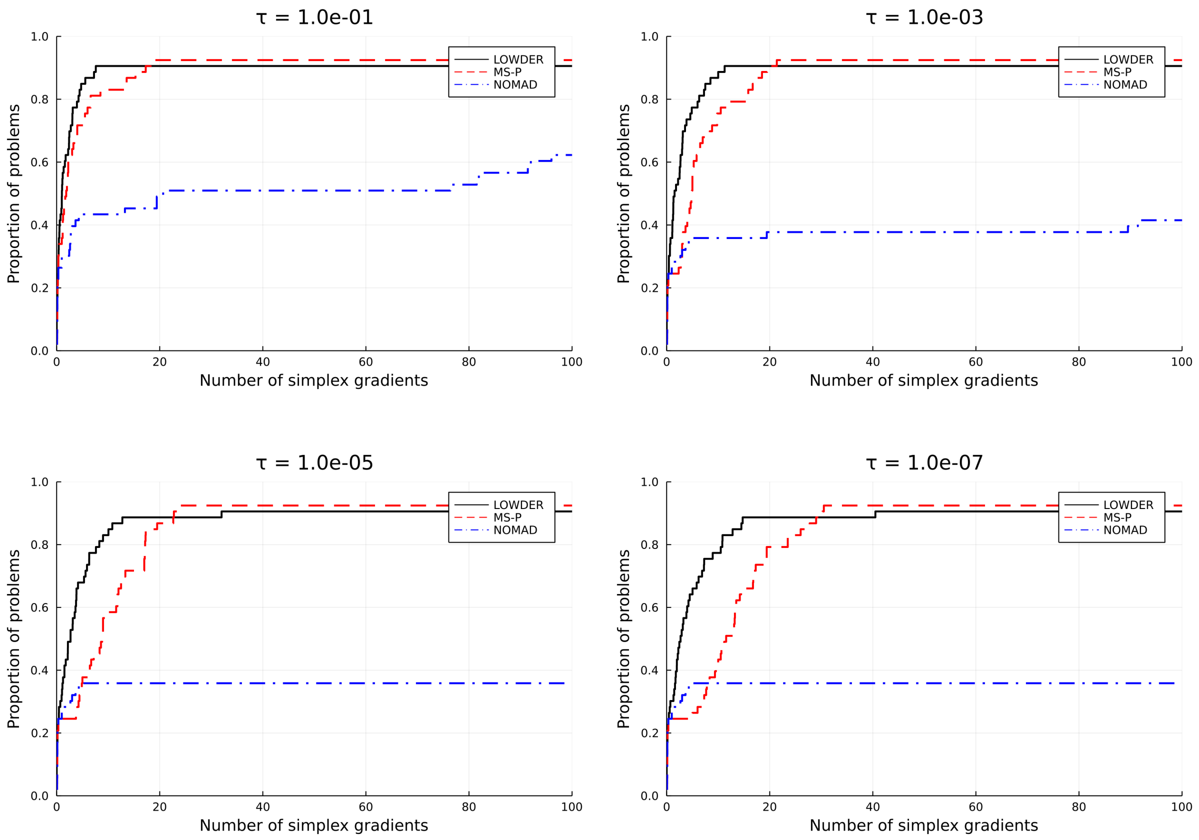


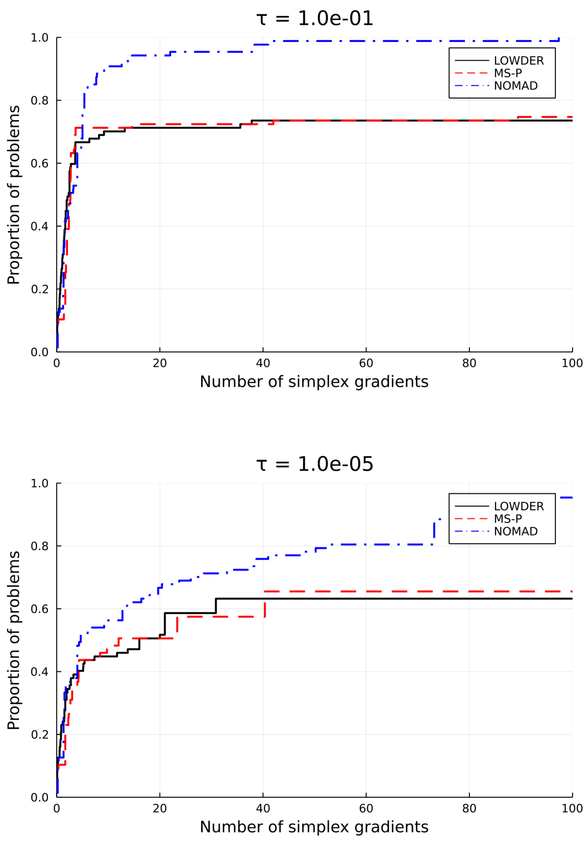
Figure 4.2: Data profiles for the problems in MW test set with tolerance $\tau \in \{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$.

4.3.2 HS test set

Analyzing Figure 4.3, we can see that `NOMAD` can solve all problems for $\tau = 10^{-1}$ and can solve 95% or more problems for other values of tolerance. `LOWDER` and `MS-P` have similar performances, alternating in the second position. When considering the computational budget of 100 simplex gradients, `LOWDER` is overcome by `MS-P` by 1% to 3%. For budget values between 20 and 40 simplex gradients, `LOWDER` performs slightly better than `MS-P`.

The great performance of `NOMAD` can be explained by the small size of the problems and by the recognized ability of the algorithm to solve complex problems, especially for non-linear problems with constraints, as is the case of this test set. Note that LOWDER employs only linear models to solve the trust-region subproblem (3.4). This can harm its performance in the case of strongly non-linear problems since the solutions generated by TRSBOX tend not to satisfy the conditions of the step acceptance phase (line 8), favoring set improvement iterations with directions calculated by ALTMOV, which may not be directions in which the objective function decrease. Another factor that can impact the quality of the solutions obtained by LOWDER is the existence of several problems in HS test set that have an infinity of local minima. NOMAD has two distinct mechanisms that help it escape from local minima. The first one is a global search procedure called SEARCH step, which can return any point on the underlying mesh, always trying to identify points that improve the best solution found so far. The second is a local search called the POOL step, and your purpose is to generate trial mesh points in the vicinity of the best solution [6]. Another behavior of the NOMAD solver that we noticed during the execution of these problems was its tendency to find local and global solutions on the boundary of the feasible set. Among the 87 problems that constitute the HS test set, NOMAD found boundary solutions for 55 of them. Considering its good performance in the data profile, this suggests the solver is very efficient when exploring and evaluating the limits of the feasible set.

When looking at the proportion of problems solved by the three solvers, we can see that they can solve about 40% of problems with 5 simplex gradients or less. From that point on, the performance differences are quite significant, especially when comparing NOMAD with the other solvers. By setting $\tau = 10^{-3}$, NOMAD can solve 60% of problems with just 13 simplex gradients, while LOWDER needs 17, and MS-P needs about 37 simplex gradients. Another situation worth mentioning occurs with $\tau = 10^{-7}$. In this case, considering the 50% mark of problems solved, NOMAD can reach this value with 6 simplex gradients, while LOWDER and MS-P need approximately 28 and 33 simplex gradients, respectively.



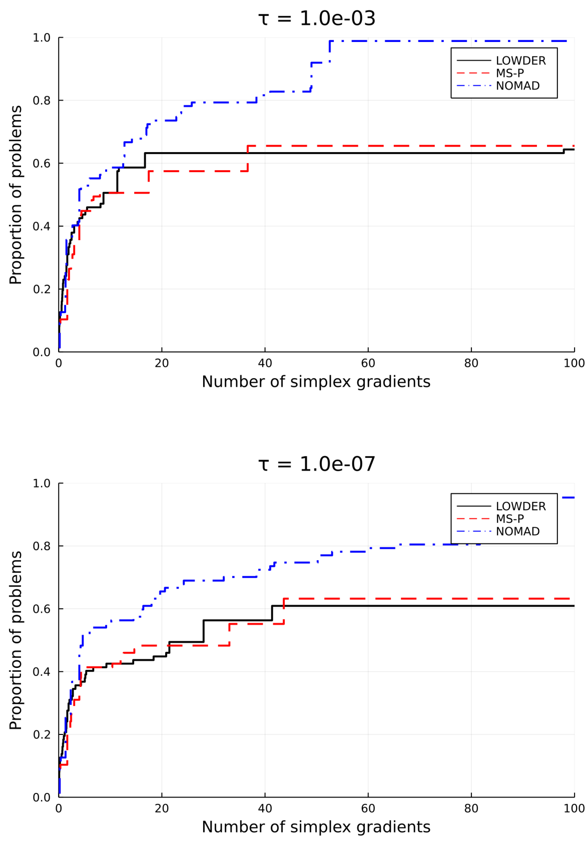


Figure 4.3: Data profiles for the problems in HS test set with tolerance $\tau \in \{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$.

4.3.3 QD test set

Our goal in this test set is to show the benefits that `LOWDER` enjoys due to exploring the structure of LOVO problems. Figure 4.4 presents the data profiles generated with tolerance $\tau = 10^{-5}$ for the problems in the QD test set, taking into account the subsets generated with 10, 25, 50, 75, and 100 component functions. Analyzing the plots presented, if we consider a computational budget of 100 simplex gradients, the robustness of the algorithms is not affected by the number of functions f_i. `LOWDER` and `NOMAD` manage to solve approximately 85% of the problems while `MS-P` solves 77% of the problems, in the 5 scenarios presented. Note that `LOWDER` outperforms `MS-P` and `NOMAD` by a large amount, especially for computational budgets of less than 40 simplex gradients. For this fixed budget, `MS-P` also outperforms `NOMAD`, but the situation reverses for higher budget values. More specifically, the data profile shows us that for robustness rates above 70%, `NOMAD` is more efficient than

MS-P and is able to match LOWDER using less than 60 simplex gradients.

Another relevant fact is that the curves presented by MS-P and NOMAD do not appear to be affected by the variation in the number of component functions, while the LOWDER data profile curve has a clear tendency to approach the vertical axis according to the number of component functions increases. That is, the performance of LOWDER tends to improve, especially when considering small computational budgets, with less than 20 simplex gradients. One way to visualize this influence is to verify the number of simplex gradients needed to solve a certain percentage of problems, as shown in Table 4.2.

Test subset	20%	40%	60%	80%	85%
QD10	5	6	7	21	34
QD25	3	3	4	14	23
QD50	2	2	3	11	19
QD75	2	2	2	10	18
QD100	1	2	2	9	17

Table 4.2: Approximate number of function simplex gradients that LOWDER needs to solve given ranges of problems on the QD test subsets.

Note that as the number of component functions increases, represented by the number associated with the test subset, there is a tendency for the number of simplex gradients to decrease, and this can be verified for all ranges of problems considered.

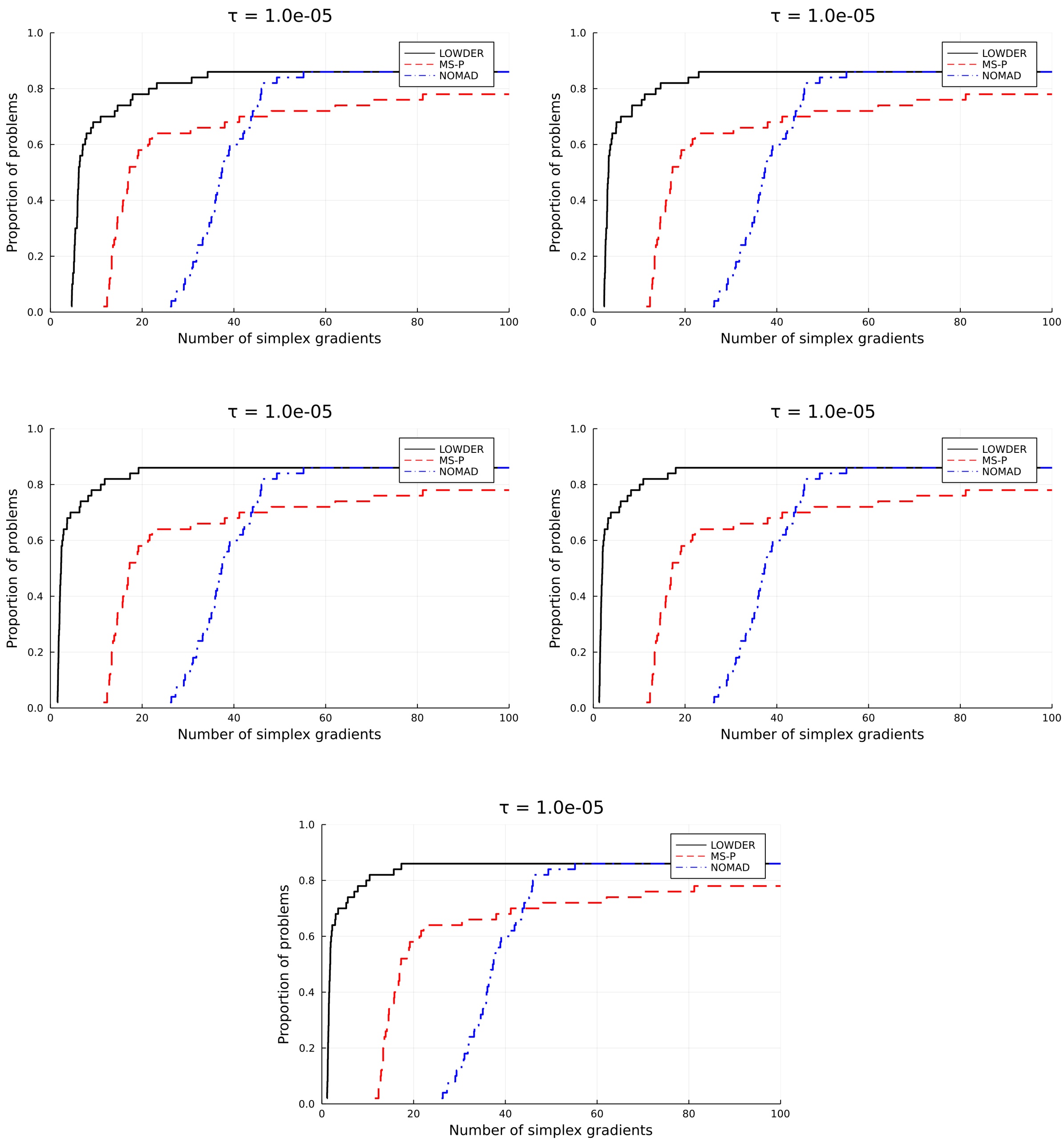


Figure 4.4: Data profiles for the problems in QD test set, with $\tau = 10^{-5}$, and $r \in \{10, 25, 50, 75, 100\}$.

Chapter 5

Conclusions and suggestions for future work

In this work, we introduced a new class of low order-value optimization methods, considering an approach reasoned on model-based derivative-free optimization. We presented a derivative-free trust-region algorithm for constrained black-box LOVO problems, which is based on the algorithms proposed by Conejo et al. [17] and Verdério et al. [72], and the ideas discussed by Andreani et al. [1]. Our algorithm can deal with general closed convex constraints and is specially designed for problems whose objective function values are provided through an oracle (black-box function). Note that we assume that we know how to project an arbitrary point onto the feasible set. Algorithm 1 has a structure very similar to the traditional trust-region framework and considers two different radii, one for the sample region and another for the trust-region. We allowed some freedom in the choice of models, as long as the gradient of the model is a good approximation of the gradient of the selected component function, in the sense of well poisedness (Assumption 3.5).

We discussed global convergence results of the algorithm, adopting common assumptions for this class of problems, as well as an interpretation from the perspective of the classical theory of LOVO problems. We also showed how the choice and construction of models can help us to obtain good theoretical properties. In this sense, we dedicated the first chapter to organize several results from the literature about error bounds for linear and quadratic models related to derivative-free trust-region algorithms. We also extended the results of [72] to underdetermined models, allowing "inexact" interpolation, and provided a clearer proof than [27] for the bound on $\left\|\mathbf{L_s}^{\dagger}\right\|$ in the minimum Frobenius norm case.

Inspired by the works of Cartis and Roberts [13] and Garmanjani et al. [35], we also studied the worst-case complexity analysis of Algorithm 1, which showed us that the number of iterations and function evaluations performed by the algorithm is in line with what is expected for model-based methods, and that the adoption of minimum Frobenius-norm quadratic models is competitive in terms of function evaluations when compared to complete determined models.

We also presented `LOWDER`, our implementation of Algorithm 1 in `Julia` language, discussed implementation details and performed numerical tests. In its current version, `LOWDER` solves bound-constrained black-box LOVO problems and builds only determined linear models. `LOWDER` has several practical improvements, many of them derived from `BOBYQA`, a general-purpose derivative-free optimization solver for bound-constrained problems. In particular, `LOWDER` inherits the initial sampling mechanisms and has simplified versions of the `TRSBOX` and `ALTMOV` routines for solving the trust region subproblem (3.4) and improving the geometry of the sample set, respectively. Like `BOBYQA`, we also avoided completely rebuilding models through an update mechanism.

Since `LOWDER` is designed for LOVO problems, we compared it with algorithms that can handle, in some way, this type of problem. In this sense, we selected `MS-P` [44] and `NOMAD` [6]. `MS-P` is a manifold sampling algorithm for composition minimization problems. `NOMAD` is a well-established algorithm for black-box optimization problems based on the direct search.

We proposed a test suite with three sets: MW, HS, and QD. MW is our main test set and contains the problems of Moré and Wild [52] for benchmarking derivative-free optimization algorithms. HS is a test set created with a combination of problems from the Hock and Schittkowski [41] collection for testing nonlinear optimization algorithms. Finally, QD is a test set created by us to measure the impact of the number of component functions on the performance of `LOWDER`. Despite not being the most robust algorithm, `LOWDER` can solve about 90% of the problems from MW with less than 20 simplex gradients of the objective function, being the most efficient algorithm for this range of computational budget. In the QD test set, `LOWDER` is the most efficient and robust algorithm. Although `NOMAD` is the least efficient algorithm for budgets smaller than 40 simplex gradients, it outperforms `MS-P` and matches LOWDER for larger budgets. Despite having similar performances for budgets of up to 5 simplex gradients in the HS test set, in general, NOMAD had the best performance and robustness, managing to solve practically all problems with a budget of 100 simplex gradients. MS-P and LOWDER obtain similar performances, solving about 70% of the problems with tolerance $\tau = 10^{-1}$ and little more than 60% of the problems considering smaller values of τ. The worse performance of LOWDER can be explained by the fact that we employ only linear models, and since the problems in HS are strongly nonlinear, this fact can impair the acceptance phases of the step and the general progress of the algorithm.

Future work may include the estimation of bounds for the Hessian of minimum norm underdetermined quadratic models. In [26], the authors obtained bounds for the projection of errors onto a specific linear subspace, which is not very useful in practical terms. On the other hand, [27, p. 79], suggests that by using an overall poisedness constant for the sample set, it is possible to establish bounds for the Hessian of the model. Another interesting question is whether models constructed by support vector regression [72] using underdetermined quadratic polynomials are able to satisfy Assumption 2.4 and their practical benefits under noisy blackbox functions.

In order to increase the performance of LOWDER, we can implement the construction of determined and underdetermined quadratic models, based on the mechanisms proposed by Powell [59] for the BOBYQA solver, and also modified versions of RESCUE, to avoid the full reconstruction of the models. In addition, we can improve the implementation of the linear models using more efficient ways to calculate and update the QR factorization. Another possible advance is the usage of the sampling strategies presented by Hough and Roberts [42] and thus avoiding situations for which Λ-poised sets are impossible to be constructed in constrained problems.

Furthermore, we can implement a mechanism of long-term memory and store relevant information about old sample points, such as objective function value, component function index, and stationarity measure. This information can be useful in constructing new sample sets and saving objective function evaluations. When interpreting LOVO as a nonsmooth composite minimization problem, such a memory mechanism can also add information about the minimum function when we calculate the function f_{min} completely, similar to what happens in the manifold sampling algorithms proposed in [44, 46].

Finally, it is well known that the Low Order-Value Optimization can generalize the nonlinear least-squares problem, as it allows us to discard observations considered outliers, as we can see in [2, 15, 31, 66]. Therefore, we can enhance `LOWDER` to take advantage of the structure of the least-squares problem, such as well-established algorithms like `DFO-GN` [13], `POUNDERS` [74], and `DFBOLS` [77], making it a competitive solver in this segment.

Bibliography

[1] Roberto Andreani, José Mario Martínez, and Leandro Martínez. Trust-region superposition methods for protein alignment. *IMA journal of numerical analysis*, 28(4):690–710, 2008.

[2] Roberto Andreani, José Mario Martínez, Leandro Martínez, and Flávio S. Yano. Low order-value optimization and applications. *Journal of Global Optimization*, 43(1):1–22, 2009.

[3] Roberto Andreani, Gabriel Haeser, and José Mario Martínez. On sequential optimality conditions for smooth constrained optimization. *Optimization*, 60(5):627–641, 2011.

[4] María B. Arouxét, Nélida Echebest, and Elvio A. Pilotta. Active-set strategy in Powell's method for optimization without derivatives. *Computational & Applied Mathematics*, 30:171–196, 2011.

[5] Charles Audet and Warren Hare. *Derivative-free and blackbox optimization*. Springer, 2017.

[6] Charles Audet, Sébastien Le Digabel, Viviane Rochon Montplaisir, and Christophe Tribes. Algorithm 1027: NOMAD version 4: Nonlinear optimization with the MADS algorithm. *Transactions on Mathematical Software*, 48(3):35:1–35:22, 2022.

[7] Manuel Berkemeier and Sebastian Peitz. Derivative-free multiobjective trust region descent method using radial basis function surrogate models. *Mathematical and Computational Applications*, 26(2):31, 2021.

[8] Stephen C. Billups, Steven P. Dirkse, and Michael C. Ferris. A comparison of large scale

mixed complementarity problem solvers. *Computational Optimization and Applications*, 7(1):3–25, 1997.

[9] Ernesto G. Birgin and Jan M. Gentil. Evaluating bound-constrained minimization software. *Computational Optimization and Applications*, 53(2):347–373, 2012.

[10] Ernesto G. Birgin, Luis F. Bueno, Natasa Krejić, and José Mario Martínez. Low order-value approach for solving var-constrained optimization problems. *Journal of Global Optimization*, 51(4):715–742, 2011.

[11] Ernesto G. Birgin, Natasa Krejić, and José Mario Martínez. Inexact restoration for derivative-free expensive function minimization and applications. *J. Comp. Appl. Math.*, 410:114193, 2022.

[12] Luis F. Bueno, Ana Friedlander, José Mario Martínez, and Francisco N. C. Sobral. Inexact restoration method for derivative-free optimization with smooth constraints. *SIAM Journal on Optimization*, 23(2):1189–1213, 2013.

[13] Coralia Cartis and Lindon Roberts. A derivative-free Gauss-Newton method. *Mathematical Programming Computation*, 11(4):631–674, 2019.

[14] Coralia Cartis and Lindon Roberts. Scalable subspace methods for derivative-free nonlinear least-squares optimization. arXiv:2102.12016, 2021.

[15] Emerson V. Castelani, Ronaldo Lopes, Wesley V. I. Shirabayashi, and Francisco N. C. Sobral. A robust method based on lovo functions for solving least squares problems. *Journal of Global Optimization*, 80(2):387–414, 2021.

[16] Frank H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

[17] Paulo D. Conejo, Elizabeth W. Karas, Lucas G. Pedroso, Ademir A. Ribeiro, and Mael Sachine. Global convergence of trust-region algorithms for convex constrained minimization without derivatives. *Applied Mathematics and Computation*, 220:324–330, 2013.

[18] Paulo D. Conejo, Elizabeth W. Karas, and Lucas G. Pedroso. A trust-region derivative-free algorithm for constrained optimization. *Optimization Methods and Software*, 30(6): 1126–1145, 2015.

[19] Andrew R. Conn and Philippe L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In *Nonlinear optimization and applications*, pages 27–47. Springer, 1996.

[20] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, 25(2):433–460, 1988.

[21] Andrew R. Conn, Nicholas I. M. Gould, Annick Sartenaer, and Philippe L. Toint. Convergence properties of minimization algorithms for convex constraints using a structured trust region. *SIAM Journal on Optimization*, 6(4):1059–1086, 1996.

[22] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization. *Mathematical Programming*, 73(1):73–110, 1996.

[23] Andrew R. Conn, Katya Scheinberg, and Philippe L. Toint. On the convergence of derivative-free methods for unconstrained optimization. *Approximation theory and optimization: tributes to MJD Powell*, pages 83–108, 1997.

[24] Andrew R. Conn, Katya Scheinberg, and Philippe L. Toint. A derivative free optimization algorithm in practice. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4718, 1998.

[25] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. *Trust-region methods.* SIAM, 2000.

[26] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation. *IMA J. Numer. Anal.*, 28(4):721–748, 2008.

[27] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to derivative-free optimization*. SIAM, Philadelphia, 2009.

[28] Andrew R. Conn, Katya Scheinberg, and Luís N. Vicente. Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points. *SIAM Journal on Optimization*, 20(1):387–415, 2009.

[29] Frank E. Curtis, Tim Mitchell, and Michael L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optimization Methods and Software*, 32(1):148–181, 2017.

[30] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

[31] Edilaine S. Duran. Uma classe de métodos do tipo Levenberg-Marquardt com passos múltiplos para problemas de Otimização de Menor Valor Ordenado. Master's thesis, Universidade Estadual de Maringá, Maringá PR BR, 2020.

[32] Giovanni Fasano, José L. Morales, and Jorge Nocedal. On the geometry phase in model-based algorithms for derivative-free optimization. *Optimization Methods & Software*, 24 (1):145–154, 2009.

[33] Priscila S. Ferreira, Elizabeth W. Karas, and Mael Sachine. A globally convergent trust-region algorithm for unconstrained derivative-free optimization. *Computational and Applied Mathematics*, 34(3):1075–1103, 2015.

[34] Priscila S. Ferreira, Elizabeth W. Karas, Mael Sachine, and Francisco N. C. Sobral. Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming. *Optimization*, 66(2):271–292, 2017.

[35] Rohollah Garmanjani, Diogo Júdice, and Luís N. Vicente. Trust-region methods without using derivatives: worst case complexity and the nonsmooth case. *SIAM Journal on Optimization*, 26(4):1987–2011, 2016.

[36] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software (TOMS)*, 29(4):373–394, 2003.

[37] Geovani N. Grapiglia, Jinyun Yuan, and Ya-xiang Yuan. A derivative-free trust-region algorithm for composite nonsmooth optimization. *Computational and Applied Mathematics*, 35(2):475–499, 2016.

[38] Serge Gratton, Philippe L. Toint, and Anke Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optimization Methods and Software*, 26(4-5):873–894, 2011.

[39] Serge Gratton, Clément W. Royer, Luís N. Vicente, and Zaikun Zhang. Complexity and global rates of trust-region methods based on probabilistic models. *IMA Journal of Numerical Analysis*, 38(3):1579–1597, 2018.

[40] Elzain A. E. Gumma, Mohsin H. A. Hashim, and M. Montaz Ali. A derivative-free algorithm for linearly constrained optimization problems. *Computational Optimization and Applications*, 57(3):599–621, 2014.

[41] Willi Hock and Klaus Schittkowski. Test examples for nonlinear programming codes. *Journal of optimization theory and applications*, 30(1):127–129, 1980.

[42] Matthew Hough and Lindon Roberts. Model-based derivative-free methods for convex-constrained optimization. *SIAM Journal on Optimization*, 32(4):2552–2579, 2022.

[43] Ching-hsiang Hung and Thomas L. Markham. The Moore-Penrose inverse of a partitioned matrix M=(ADBC). *Linear Algebra and its Applications*, 11(1):73–86, 1975.

[44] Jeffrey Larson and Matt Menickelly. Structure-aware methods for expensive derivative-free nonsmooth composite optimization. arXiv:2207.08264, 2022.

[45] Jeffrey Larson, Matt Menickelly, and Stefan M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[46] Jeffrey Larson, Matt Menickelly, and Baoyu Zhou. Manifold sampling for optimizing nonsmooth nonconvex compositions. *SIAM Journal on Optimization*, 31(4):2638–2664, 2021.

[47] José Mario Martínez. Order-value optimization and new applications. In *ICIAM 07: 6th International Conference on Industrial and Applied Mathematics, Zürich, Switzerland, 16-20 July 2007: Invited Lectures*, pages 279–296. European Mathematical Society, 2009.

[48] José Mario Martínez. Generalized order-value optimization. *Top*, 20(1):75–98, 2012.

[49] Leandro Martínez, Roberto Andreani, and José Mario Martínez. Convergent algorithms for protein structural alignment. *BMC bioinformatics*, 8(1):1–15, 2007.

[50] Hans D. Mittelmann. Benchmarking interior point LP/QP solvers. *Optimization Methods and Software*, 11(1-4):655–670, 1999.

[51] Alexis Montoison, Pierrick Pascal, and Ludovic Salomon. NOMAD.jl: A Julia interface for the constrained blackbox solver NOMAD. `https://github.com/bbopt/NOMAD.jl`, July 2020.

[52] Jorge J. Moré and Stefan M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[53] Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.

[54] John A. Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[55] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[56] Bruna L. Pelegrini, Frederico M. B. Fernandes, Thiago Fernandes, et al. Novel green strategy to improve the hydrophobicity of cellulose nanocrystals and the interfacial elasticity of pickering emulsions. *Cellulose*, 28(10):6201–6238, 2021.

[57] Michael J. D. Powell. UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582, 2002.

[58] Michael J. D. Powell. The NEWUOA software for unconstrained optimization without derivatives. In *Large-scale nonlinear optimization*, pages 255–297. Springer, Boston, 2006.

[59] Michael J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pages 26–46, 2009.

[60] Michael J. D. Powell. On the convergence of trust region algorithms for unconstrained minimization without derivatives. *Computational Optimization and Applications*, 53(2): 527–555, 2012.

[61] Michael J. D. Powell. On fast trust region methods for quadratic models with linear constraints. *Mathematical Programming Computation*, 7(3):237–267, 2015.

[62] Ademir A. Ribeiro and Elizabeth W. Karas. *Otimização contínua: Aspectos Teóricos e Computacionais*. Cengage Learning, 2013.

[63] Luis M. Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[64] Jong-Hyun Ryu and Sujin Kim. A derivative-free trust-region method for biobjective optimization. *SIAM Journal on Optimization*, 24(1):334–362, 2014.

[65] Katya Scheinberg and Philippe L. Toint. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization*, 20(6):3512–3532, 2010.

[66] Anderson E. Schwertner. O método de Levenberg-Marquardt para problemas de otimização de menor valor ordenado. Master's thesis, Universidade Estadual de Maringá, Maringá PR BR, 2019.

[67] Anderson E. Schwertner and Francisco N. C. Sobral. On complexity constants of linear and quadratic models for derivative-free trust-region algorithms. arXiv:2205.11358, 2022.

[68] Wiliam Spendley, George R. Hext, and Francis R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4): 441–461, 1962.

[69] Jana Thomann and Gabriele Eichfelder. A trust-region algorithm for heterogeneous multiobjective optimization. *SIAM Journal on Optimization*, 29(2):1017–1047, 2019.

[70] Anke Tröltzsch. *An active-set trust-region method for bound-constrained nonlinear optimization without derivatives applied to noisy aerodynamic design problems*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2011.

[71] Anke Tröltzsch, Caslav Ilic, and Martin Siggel. SQPDFO-a Trust-Region Based Algorithm for Generally-Constrained Derivative-Free Optimization. *Proceedings of the 13th AMiTaNS*, 2021.

[72] Adriano Verdério, Elizabeth W. Karas, Lucas G. Pedroso, and Katya Scheinberg. On the construction of quadratic models for derivative-free trust-region algorithms. *EURO J. Comput. Optim.*, 5(4):501–527, 2017.

[73] Stefan M. Wild. *Derivative-free optimization algorithms for computationally expensive functions*. PhD thesis, Cornell University, Ithaca, 2008.

[74] Stefan M. Wild. *POUNDERS in TAO: Solving Derivative-Free Nonlinear Least-Squares Problems with POUNDERS*, chapter 40, pages 529–539. MOS-SIAM Series on Optimization. SIAM, 2017.

[75] Stefan M. Wild, Rommel G. Regis, and Christine A. Shoemaker. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30(6):3197–3219, 2008.

[76] David H. Winfield. *Function and Functional Minimization by Interpolation in Data Tables*. PhD thesis, Harvard University, Cambridge MA USA, 1969.

[77] Hongchao Zhang, Andrew R. Conn, and Katya Scheinberg. A derivative-free algorithm for least-squares minimization. *SIAM Journal on Optimization*, 20(6):3555–3576, 2010.

Appendix A

Hock-Schittkowski selected problems

Prob.	**Objective function**	**Constraints**	**Initial guess**
1	$f_1(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$-1.5 \le x_2$	$[-2, 1]$
3	$f_3(x) = x_2 + 10^{-5}(x_2 - x_1)^2$	$0 \le x_2$	$[10, 1]$
4	$f_4(x) = \frac{1}{3}(x_1 + 1)^3 + x_2$	$1 \le x_1$ $0 \le x_2$	$[1.125, 0.125]$
5	$f_5(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$	$-1.5 \le x_1 \le 4$ $-3 \le x_2 \le 3$	$[0, 0]$
25	$f_{25}(x) = \sum_{i=1}^{99} (g_i(x))^2$, where $g_i(x) = -0.01i + \exp\big(-x_1^{-1}(u_i - x_2)^{x_3}\big)$, and $u_i = 25 + (-50\ln(0.01i))^{\frac{2}{3}}$, for $i = 1, \dots, 99$	$0.1 \le x_1 \le 100$ $0 \le x_2 \le 25.6$ $0 \le x_3 \le 5$	$[100, 12.5, 3]$
38	$f_{38}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2$ $+(1 - x_3)^2 + 10.1\big((x_2 - 1)^2 + (x_4 - 1)^2\big)$ $+19.8(x_2 - 1)(x_4 - 1)$	$-10 \le x_i \le 10$, for $i = 1, \dots, 4$	$[-3, -1, -3, -1]$
45	$f_{45}(x) = 2 - \frac{1}{120}x_1x_2x_3x_4x_5$	$0 \le x_1 \le i$, for $i = 1, \dots, 5$	$[2, 2, 2, 2, 2]$
110	$f_{110}(x) = \sum_{i=1}^{10} \Big(\ln(x_i - 2)^2 + \ln(10 - x_i)^2\Big) - \left(\prod_{j=1}^{10} x_i\right)^{0.2}$	$2.001 \le x_i \le 9.999$, for $i = 1, \dots, 10$	$[9, \dots, 9]$

Table A.1: Selected problems from Hock and Schittkowski collection, see [41].

Appendix B

HS test set problems

Prob.	**Dim.**	**Comp. functions**	**Lower bound**	**Upper bound**	**Initial guess**
1	2	f_1, f_3	$]-\infty, 0]$	$]+\infty, +\infty[$	$[-2, 1]$
2	2	f_1, f_4	$[1, 0]$	$]+\infty, +\infty[$	$[1.125, 0.125]$
3	2	f_1, f_5	$[-1.5, -1.5]$	$[4, 3]$	$[0, 0]$
4	3	f_1, f_{25}	$[0.1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
5	4	f_1, f_{38}	$[-10, -1.5, -10, -10]$	$[10, 10, 10, 10]$	$[-3, -1, -3, -1]$
6	5	f_1, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
7	10	f_1, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
8	2	f_3, f_4	$[1, 0]$	$]+\infty, +\infty[$	$[1.125, 0.125]$
9	2	f_3, f_5	$[-1.5, 0]$	$[4, 3]$	$[0, 0]$
10	3	f_3, f_{25}	$[0.1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
11	4	f_3, f_{38}	$[-10, 0, -10, -10]$	$[10, 10, 10, 10]$	$[-3, -1, -3, -1]$
12	5	f_3, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
13	10	f_3, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
14	2	f_4, f_5	$[1, 0]$	$[4, 3]$	$[1.125, 0.125]$
15	3	f_4, f_{25}	$[0.1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
16	4	f_4, f_{38}	$[1, 0, -10, -10]$	$[10, 10, 10, 10]$	$[3, 1, -3, -1]$
17	10	f_4, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
18	4	f_5, f_{38}	$[-1.5, -3, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
19	5	f_5, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
20	4	f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$

Table B.1: Problems 1 to 20 of HS test set with f_i, $i \in \mathcal{I}$, described in Table A.1.

Prob.	Dim.	Comp. functions	Lower bound	Upper bound	Initial guess
21	5	f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
22	5	f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
23	10	f_{38}, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
24	2	f_1, f_3, f_4	$[1, 0]$	$]+\infty, +\infty[$	$[10, 1]$
25	2	f_1, f_3, f_5	$[-1.5, 0]$	$[4, 3]$	$[10, 1]$
26	3	f_1, f_3, f_{25}	$[0.1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
27	4	f_1, f_3, f_{38}	$[-10, 0, -10, -10]$	$[10, 10, 10, 10]$	$[-3, -1, -3, -1]$
28	5	f_1, f_3, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
29	10	f_1, f_3, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
30	2	f_1, f_4, f_5	$[1, 0]$	$[4, 3]$	$[1.125, 0.125]$
31	3	f_1, f_4, f_{25}	$[1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
32	4	f_1, f_4, f_{38}	$[1, 0, -10, -10]$	$[10, 10, 10, 10]$	$[3, 1, -3, -1]$
33	10	f_1, f_4, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
34	4	f_1, f_5, f_{38}	$[-1.5, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
35	5	f_1, f_5, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
36	4	f_1, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
37	5	f_1, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
38	5	f_1, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
39	10	f_1, f_{38}, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
40	2	f_3, f_4, f_5	$[1, 0]$	$[4, 3]$	$[1.125, 0.125]$
41	3	f_3, f_4, f_{25}	$[1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
42	4	f_3, f_4, f_{38}	$[1, 0, -10, -10]$	$[10, 10, 10, 10]$	$[3, 1, -3, -1]$
43	10	f_3, f_4, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
44	4	f_3, f_5, f_{38}	$[-1.5, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
45	5	f_3, f_5, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
46	4	f_3, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
47	5	f_3, f_{25}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
48	5	f_3, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
49	10	f_3, f_{38}, f_{110}	$[2.001, \ldots, 2.001]$	$[9.999, \ldots, 9.999]$	$[9, \ldots, 9]$
50	3	f_4, f_5, f_{25}	$[1, 0, 0]$	$[4, 3, 5]$	$[100, 12.5, 3]$

Table B.2: Problems 21 to 50 of HS test set with f_i, $i \in \mathcal{I}$, described in Table A.1.

Prob.	**Dim.**	**Comp. functions**	**Lower bound**	**Upper bound**	**Initial guess**
51	4	f_4, f_5, f_{38}	$[1, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
52	5	f_4, f_5, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
53	4	f_4, f_{25}, f_{38}	$[1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
54	5	f_4, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
55	5	f_4, f_{38}, f_{110}	$[2.001, \dots, 2.001]$	$[9.999, \dots, 9.999]$	$[9, \dots, 9]$
56	4	f_5, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[4, 3, 5, 10]$	$[-3, -1, -3, -1]$
57	5	f_5, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
58	5	f_5, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
59	5	f_{25}, f_{38}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
60	2	f_1, f_3, f_4, f_5	$[1, 0]$	$[4, 3]$	$[1.125, 0.125]$
61	3	f_1, f_3, f_4, f_{25}	$[1, 0, 0]$	$[100, 25.6, 5]$	$[100, 12.5, 3]$
62	4	f_1, f_3, f_4, f_{38}	$[1, 0, -10, -10]$	$[10, 10, 10, 10]$	$[3, 1, -3, -1]$
63	10	f_1, f_3, f_4, f_{110}	$[2.001, \dots, 2.001]$	$[9.999, \dots, 9.999]$	$[9, \dots, 9]$
64	3	f_1, f_3, f_5, f_{25}	$[0.1, 0, 0]$	$[4, 3, 5]$	$[100, 12.5, 3]$
65	4	f_1, f_3, f_5, f_{38}	$[-1.5, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
66	5	f_1, f_3, f_5, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
67	4	f_1, f_3, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
68	5	f_1, f_3, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
69	5	f_1, f_3, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
70	10	$f_1, f_3, f_{38}, f_{110}$	$[2.001, \dots, 2.001]$	$[9.999, \dots, 9.999]$	$[9, \dots, 9]$
71	3	f_1, f_4, f_5, f_{25}	$[1, 0, 0]$	$[4, 3, 5]$	$[100, 12.5, 3]$
72	4	f_1, f_4, f_5, f_{38}	$[1, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$
73	4	f_1, f_4, f_{25}, f_{38}	$[1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
74	10	$f_1, f_4, f_{38}, f_{110}$	$[2.001, \dots, 2.001]$	$[9.999, \dots, 9.999]$	$[9, \dots, 9]$
75	4	f_1, f_5, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[4, 3, 5, 10]$	$[-3, -1, -3, -1]$
76	5	f_1, f_5, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
77	5	f_1, f_5, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
78	5	$f_1, f_{25}, f_{38}, f_{45}$	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
79	3	f_3, f_4, f_5, f_{25}	$[1, 0, 0]$	$[4, 3, 5]$	$[100, 12.5, 3]$
80	4	f_3, f_4, f_5, f_{38}	$[1, 0, -10, -10]$	$[4, 3, 10, 10]$	$[-3, -1, -3, -1]$

Table B.3: Problems 51 to 80 of HS test set with f_i, $i \in \mathcal{I}$, described in Table A.1.

Prob.	**Dim.**	**Comp. functions**	**Lower bound**	**Upper bound**	**Initial guess**
81	4	f_3, f_4, f_{25}, f_{38}	$[1, 0, 0, -10]$	$[10, 10, 5, 10]$	$[-3, -1, -3, -1]$
82	4	f_3, f_5, f_{25}, f_{38}	$[0.1, 0, 0, -10]$	$[4, 3, 5, 10]$	$[-3, -1, -3, -1]$
83	5	f_3, f_5, f_{25}, f_{45}	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
84	5	f_3, f_5, f_{38}, f_{45}	$[0, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
85	5	$f_3, f_{25}, f_{38}, f_{45}$	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$
86	4	f_4, f_5, f_{25}, f_{38}	$[1, 0, 0, -10]$	$[4, 3, 5, 10]$	$[-3, -1, -3, -1]$
87	5	$f_5, f_{25}, f_{38}, f_{45}$	$[0.1, 0, 0, 0, 0]$	$[1, 2, 3, 4, 5]$	$[2, 2, 2, 2, 2]$

Table B.4: Problems 81 to 87 of HS test set with f_i, $i \in \mathcal{I}$, described in Table A.1.

Appendix C

Data profiles for MW test set

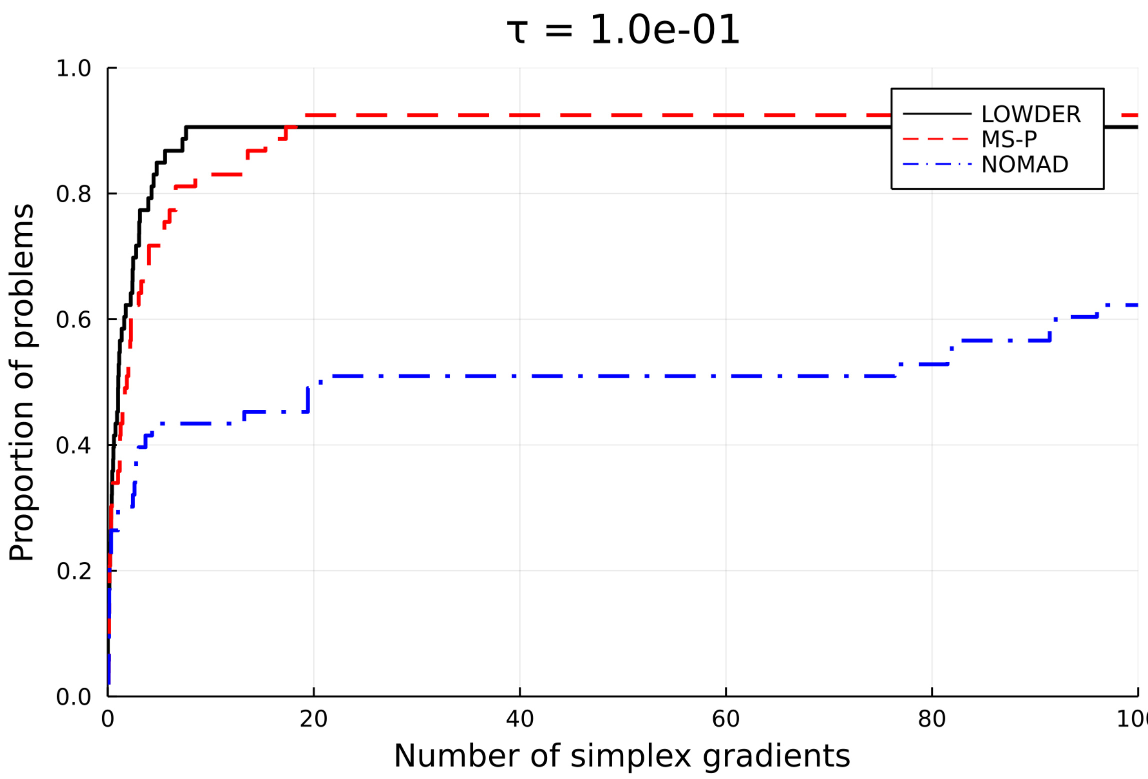


Figure C.1: Data profile for MW test set with tolerance $\tau = 10^{-1}$.

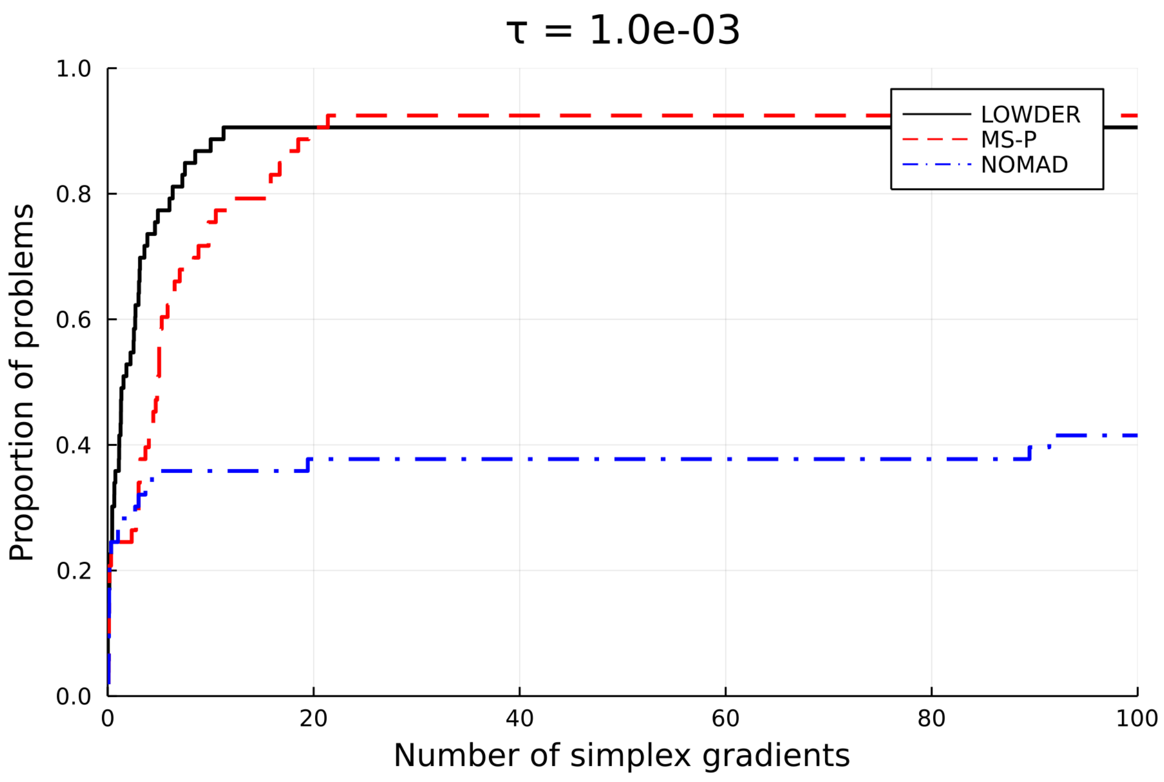


Figure C.2: Data profile for MW test set with tolerance $\tau = 10^{-3}$.

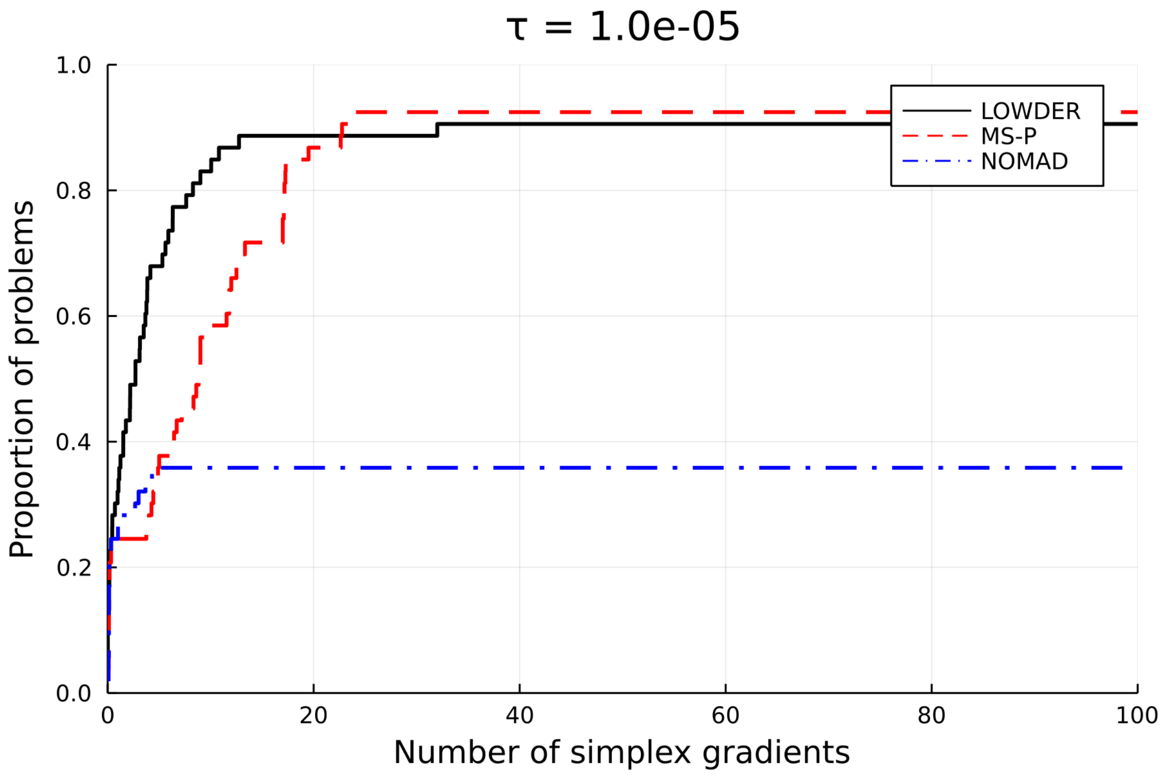


Figure C.3: Data profile for MW test set with tolerance $\tau = 10^{-5}$.

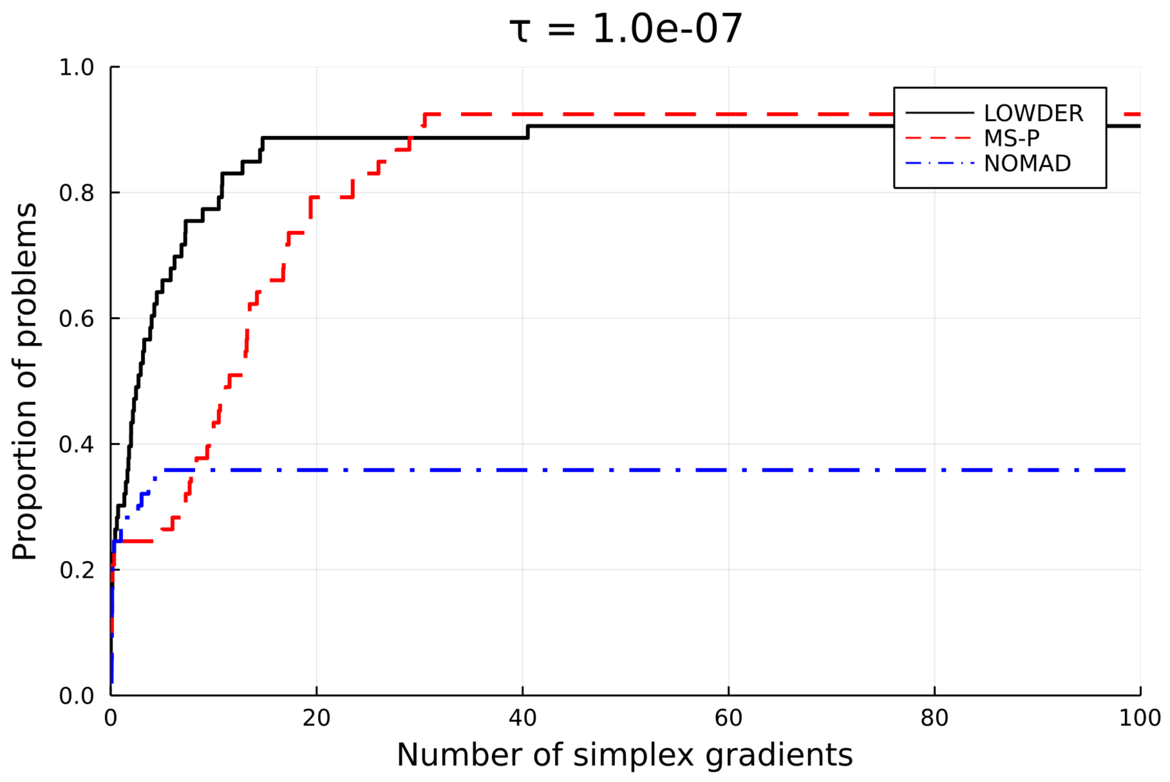


Figure C.4: Data profile for MW test set with tolerance $\tau = 10^{-7}$.

Appendix D

Data profiles for HS test set

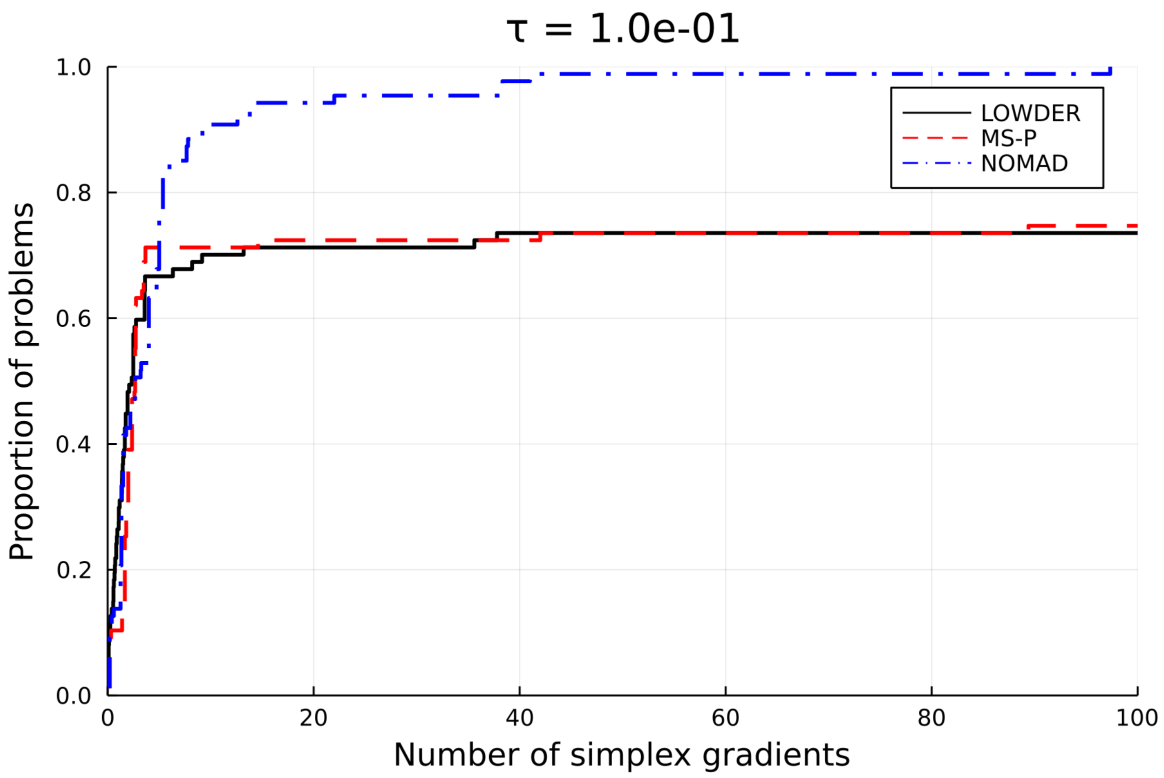


Figure D.1: Data profile for HS test set with tolerance $\tau = 10^{-1}$.

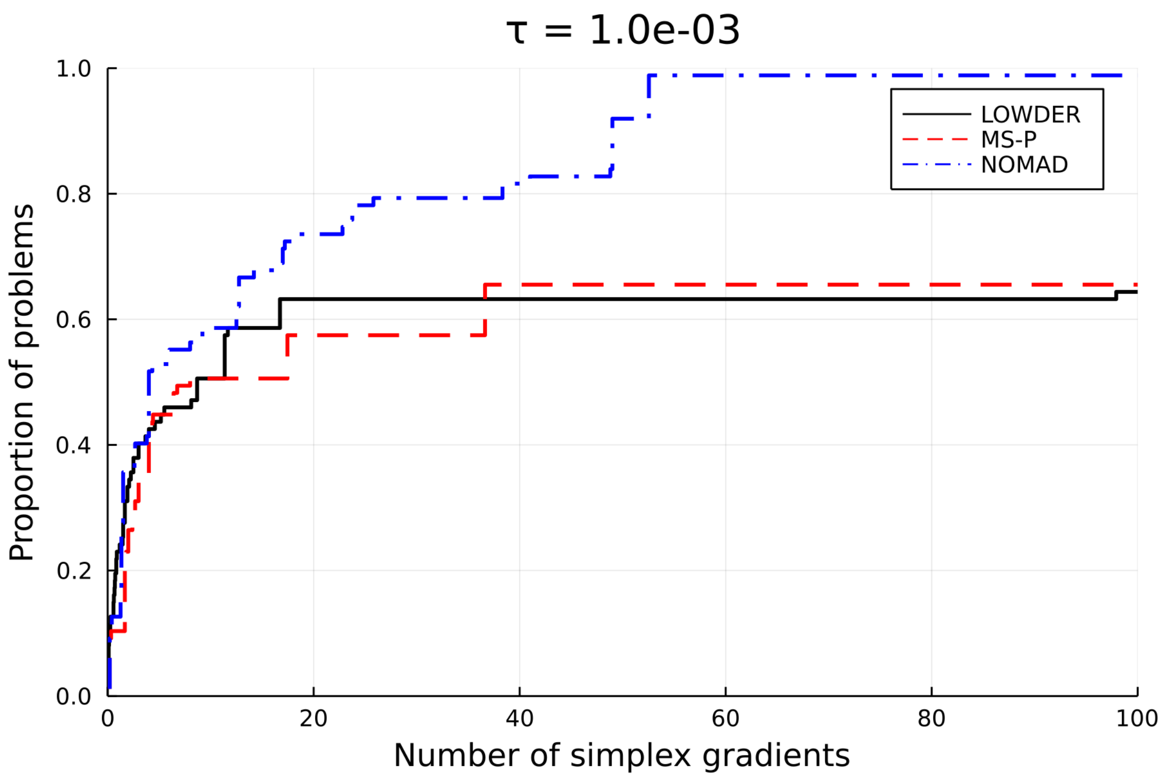


Figure D.2: Data profile for HS test set with tolerance $\tau = 10^{-3}$.

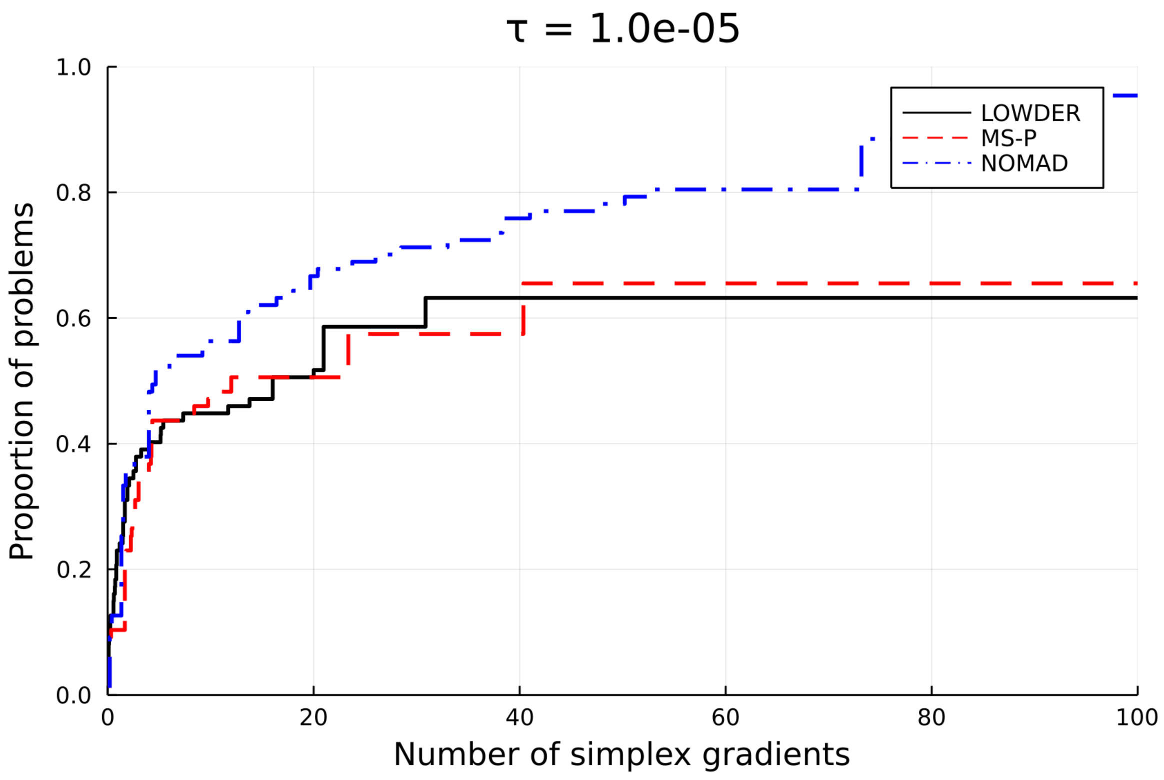


Figure D.3: Data profile for HS test set with tolerance $\tau = 10^{-5}$.

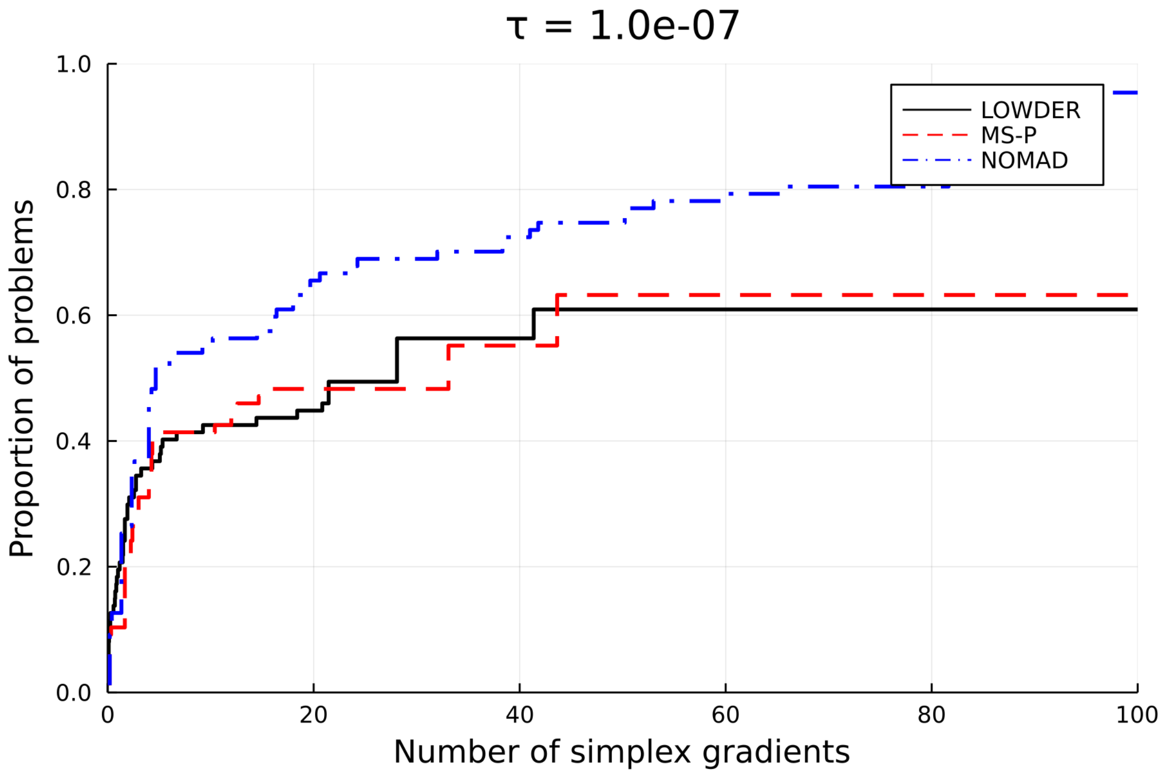


Figure D.4: Data profile for HS test set with tolerance $\tau = 10^{-7}$.

Data profiles for QD test set

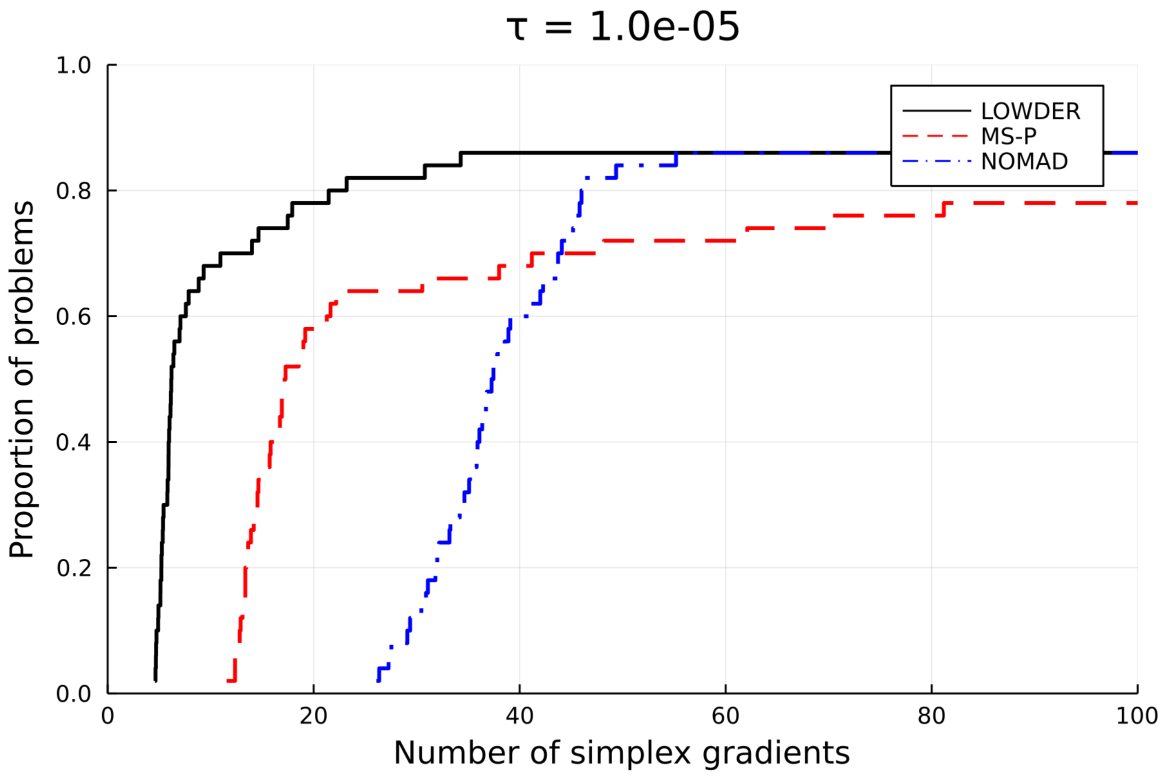


Figure E.1: Data profile for QD10 test subset with tolerance $\tau = 10^{-5}$.

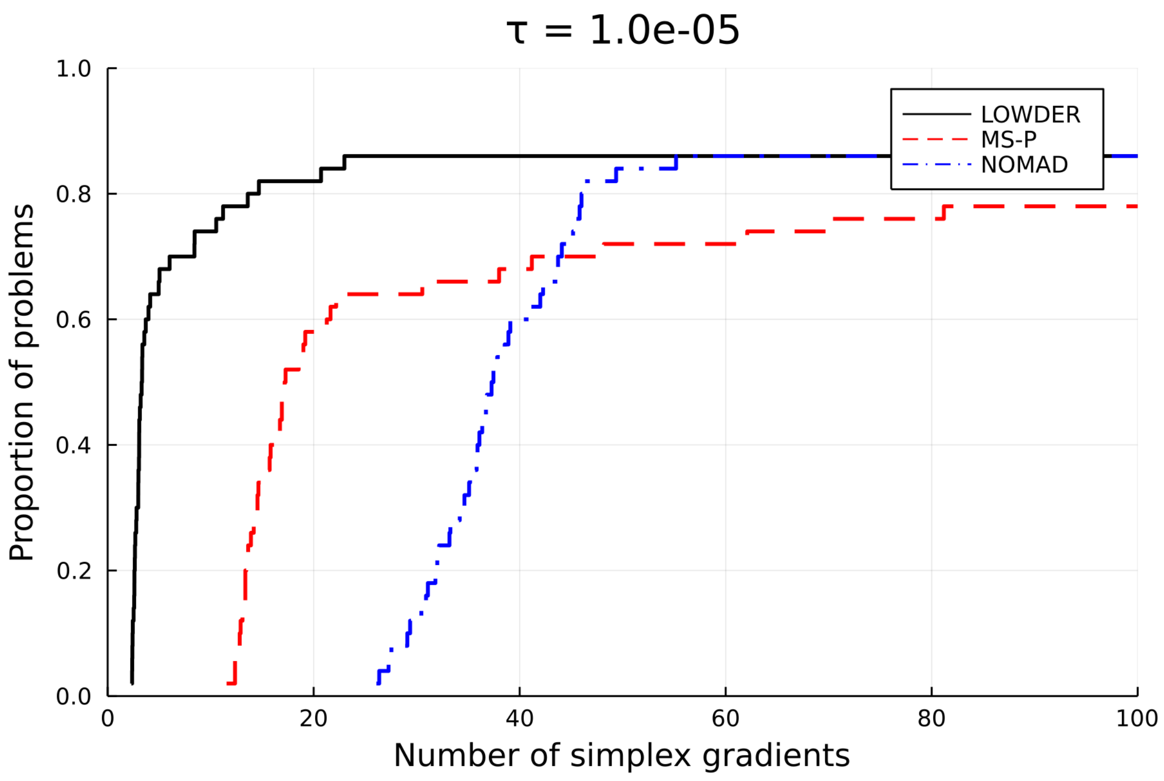


Figure E.2: Data profile for QD25 test subset with tolerance $\tau = 10^{-5}$.

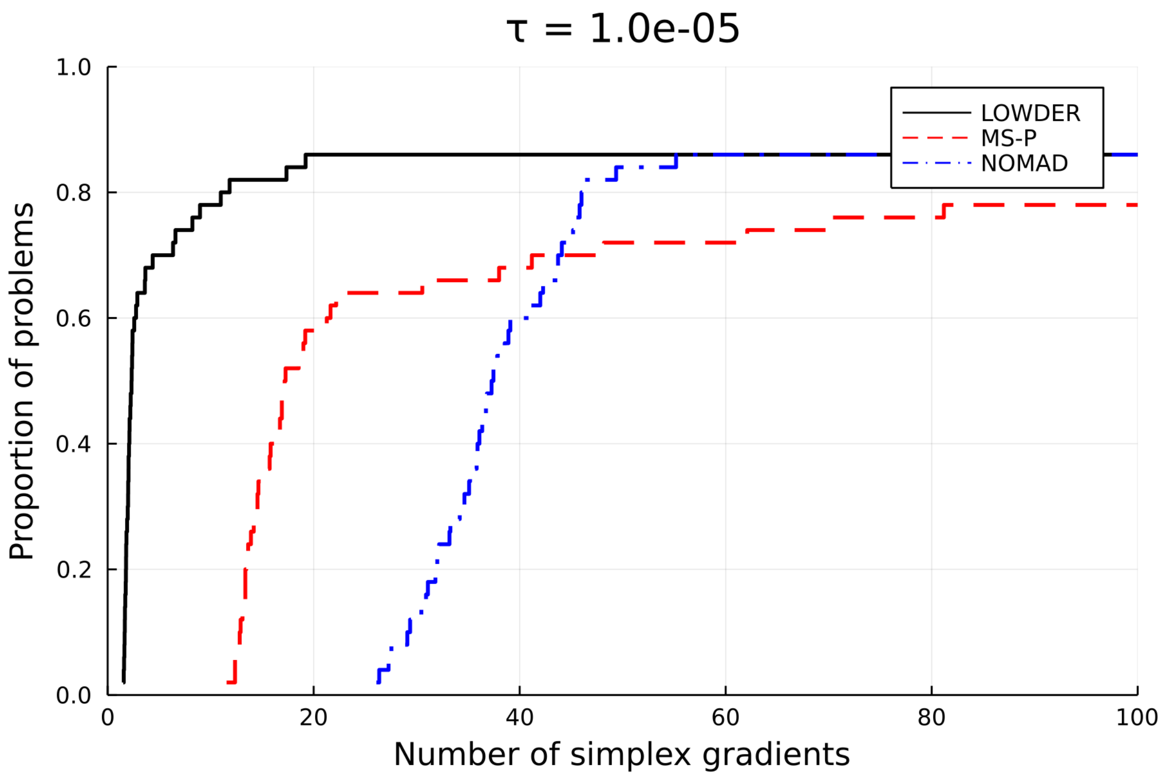


Figure E.3: Data profile for QD50 test subset with tolerance $\tau = 10^{-5}$.

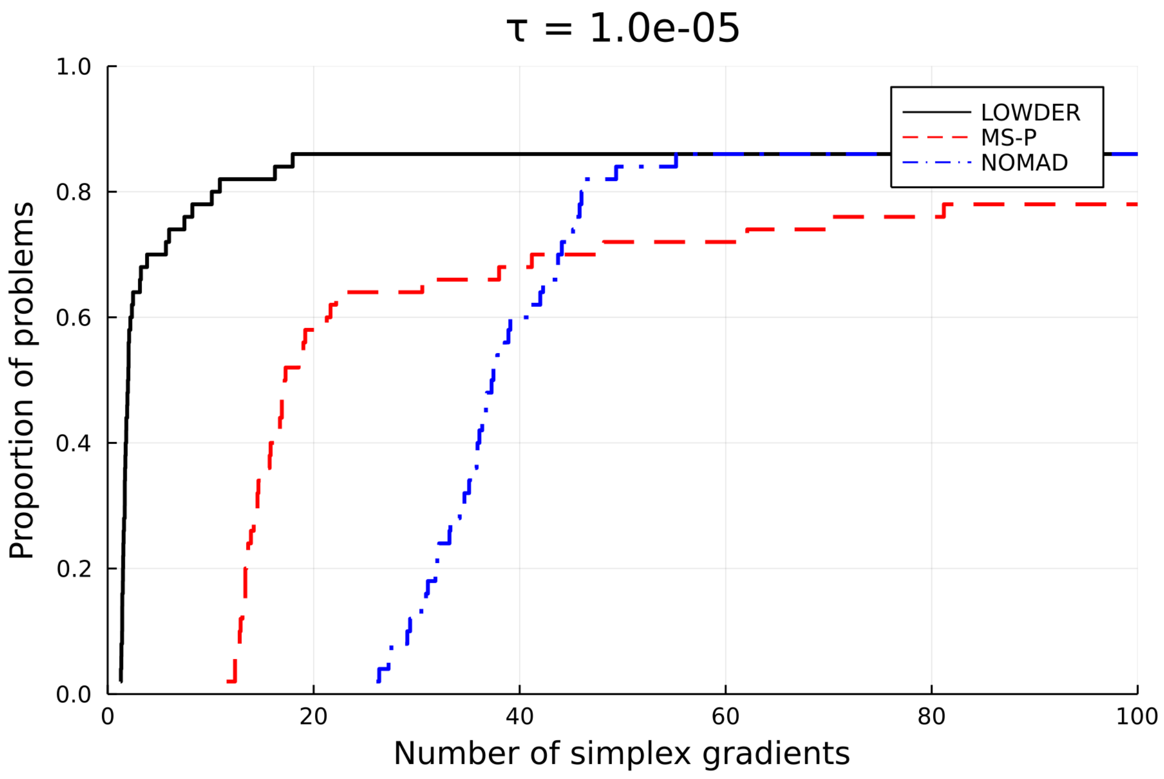


Figure E.4: Data profile for QD75 test subset with tolerance $\tau = 10^{-5}$.

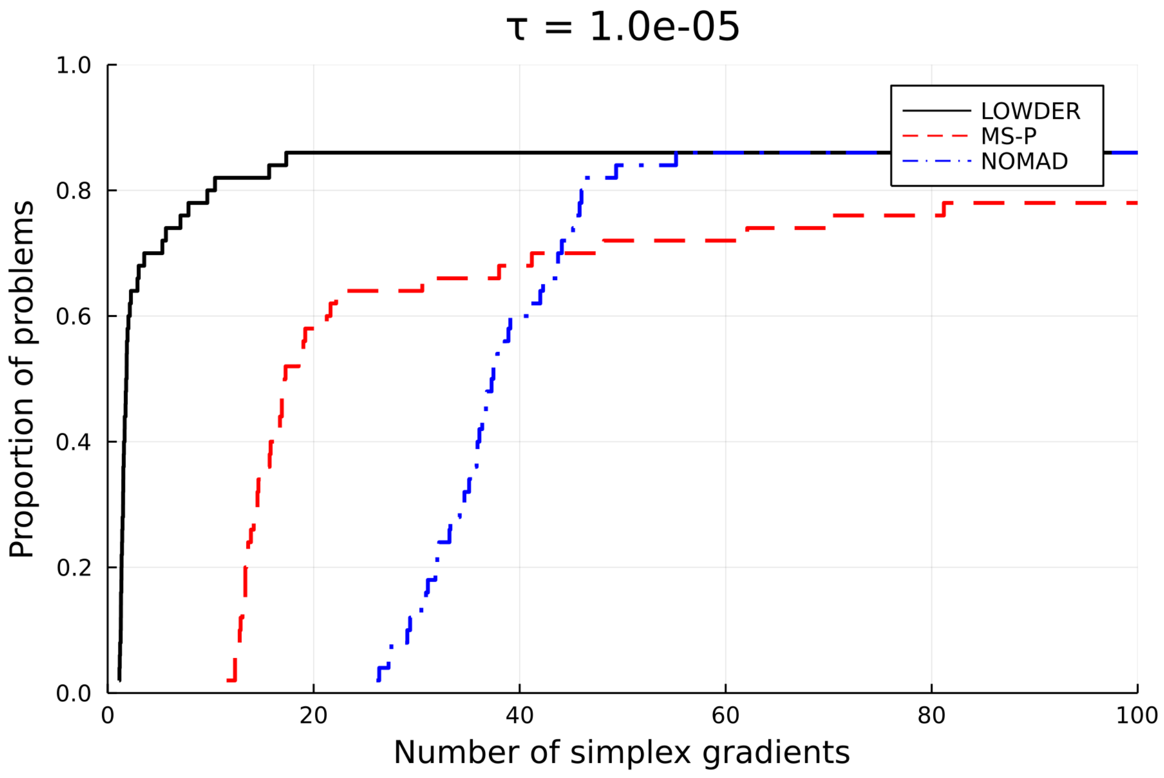


Figure E.5: Data profile for QD100 test subset with tolerance $\tau = 10^{-5}$.